JEDE AUSGABE MIT FIRMENREGISTER UND PRAKTIKUMSBÖRSE

01/2012 € 6,90

making games MAGAZIN

MAGAZIN FÜR SPIELE-ENTWICKLUNG UND BUSINESS-DEVELOPMENT

AUF DER SUCHE NACH DEM SPIELSPASS

MEHR ALS NUR EIN BAUCHGEFÜHL: WIE SICH SPASS ERKLÄREN, ENTDECKEN, ANALYSIEREN, KALKULIEREN UND ERFORSCHEN LÄSST.

4 197050 406909 01

BUSINESS
TRACKEN UND AUSWERTEN VON MOBILE-GAMES-KENNZAHLEN

PROGRAMMIERUNG
JAGGED ALLIANCE ONLINE: EIN GROSSPROJEKT MIT UNITY 3D

PROJEKTMANAGEMENT
DIE ZEHN GRÖSSTEN IRRTÜMER ÜBER DIE PRE-PRODUCTION

BILDUNG, EIN LEBEN LANG.

HARD SKILL-TRAININGS

BUSINESS TO BUSINESS

KOMPETENZMANAGEMENT

VIDEO TUTORIALS

SOFT SKILL-TRAININGS

ÜBUNGEN UND TESTS

TEAM-ENTWICKLUNG

E-LEARNING

ONLINE CAMPUS

START AB 2012

MIT UNS

INS JAHR 2012

ONLINE GAME DEVELOPMENT

AUSBILDUNG

ONLINE GAME GRAPHICS

ONLINE GAME ENGINEERING

START JEWEILS IM APRIL UND OKTOBER

CREATING MOBILE GAMES WORKSHOP

ITEM SELLING WORKSHOP

WEITERBILDUNG

UNITY 3D ENGINE WORKSHOP

CONCEPT ART WORKSHOP

KOSTENLOSES INFOPAKET UNTER **WWW.SCHOOL4GAMES.NET**
ODER DIREKT PERSÖNLICH UNTER **TEL. +49 (0)30 / 965 952 44**

S4G SCHOOL FOR GAMES GMBH - ANGEZEIGTE ERGÄNZUNGSSCHULE - GUBENER STR. 47, 10243 BERLIN

Start Making Games
Viel Spaß mit 2012

Der Autor dieser Zeilen wäre so gern ein großes, haariges Monster gewesen, mit zwei Hörnern auf dem Schädel. Stattdessen wurde er ein grünes, dreiäugiges Knubbelvieh namens »Blob«. Monster World, eines der derzeit erfolgreichsten deutschen Facebook-Spiele, hat ihm schlichtweg keine Wahl gelassen. Und warum? Weil A/B-Tests des Berliner Entwicklers wooga ergeben haben, dass deutlich mehr Spieler das Tutorial abschließen, wenn es zu Partiebeginn keine Monsterwahl gibt. Blöd. Aber es ist eben eine Entscheidung der Masse, quasi gelebte Demokratie in der Games-Entwicklung. Und ein ebenso erstaunliches wie spannendes Beispiel dafür, wie heutzutage Spielspaß analysiert, seziert und definiert wird. Die subjektiven Empfindungen einzelner – egal ob nun Spieler, Making-Games-Redakteur oder Game Designer – werden dabei immer irrelevanter. Spielspaß wird inzwischen wissenschaftlich untersucht (Seite 12), im GamesLab erforscht (Seite 14), anhand von Metriken analysiert (Seite 20) oder sogar mit Hilfe mathematischer Formeln berechnet (Seite 28). Trotzdem oder gerade deshalb ist es irgendwie beruhigend zu lesen, dass nach wie vor auch Gefühle wie Stolz, Neugier und Ordnungsliebe (!) darüber entscheiden, ob uns ein Spiel Spaß macht oder eben nicht. (Seite 25)

Das Making Games Jahr 2012

Auch ganz ohne Metriken und A/B-Tests sind wir uns ziemlich sicher, dass 2012 ein spaßiges Jahr für Making Games wird. Wir haben einiges vor, wollen unser Magazin weiter verbessern, unser Branchenverzeichnis Key Players international noch bekannter machen (Wie? Seite 11!), endlich einen Newsletter-Service anbieten sowie mit unseren Recruiting-Events Making Games Talents und Jobs & Karriere noch mehr Jobs vermitteln als im letzten Jahr (und da waren es schon über 100). Falls Sie mitmachen wollen, finden Sie hier bereits ein paar wissenswerte Termine:

8. März: German Key Players Night auf der GDC
30. März: Ende der Early-Bird-Phase für Key Players
30. April: Ende der Early-Bird-Phase für gamescom Jobs & Karriere

Und damit Sie sich diese Daten gleich irgendwo rot anstreichen können, haben wir den passenden Kalender schon mal beigelegt.

Wir wünschen Ihnen ein spannendes, kreatives und erfolgreiches 2012!

das Team von Making Games

Heiko Klinge

ist Chefredakteur vom Making Games Magazin.

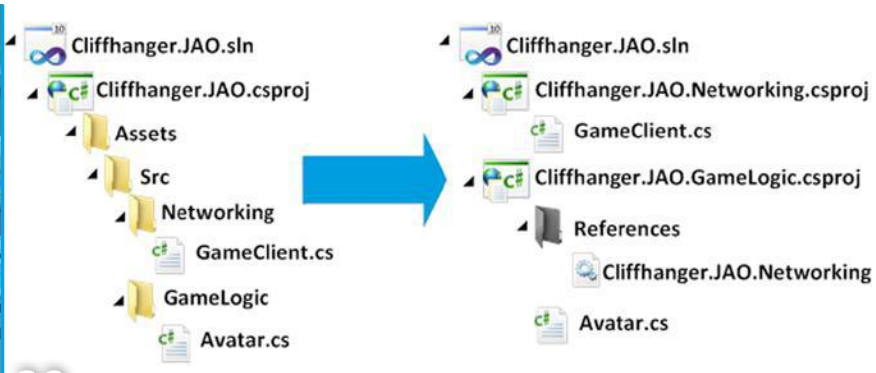
WWW
makinggames.de

- Immer wenn Sie diesen Hinweis bei einem Artikel sehen, möchten wir Sie auf ergänzende oder vertiefende Informationsangebote unseres Online-Portals aufmerksam machen.

»Spielspaß wird inzwischen wissenschaftlich untersucht, im GamesLab erforscht, anhand von Metriken analyisert oder sogar anhand mathematischer Formeln berechnet.«

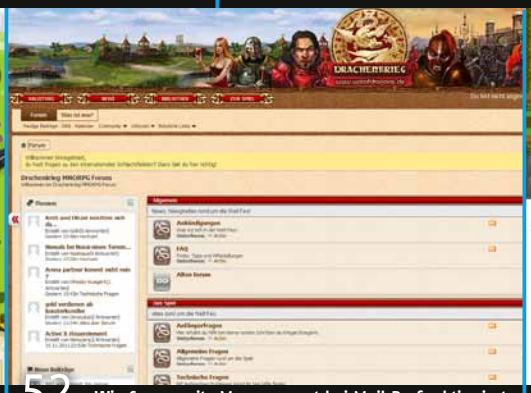
01/2012
Inhalt

56 Playfish-CEO Kristian Segerstrale über die Entstehung von Sims Social.



32 Ein Großprojekt mit Unity 3D – Johannes Scharl rekapituliert den Entwicklungsprozess.

20 Zahlenanalyse von woogas Monster World.



52 Wie Community Management bei Mail.Ru funktioniert.

38 Evolution des ältesten Multiplattform-MMOs TibiaME.

Branche

06 **Branche / Köpfe**
Bemerkenswertes aus der Branche

07 **Eventkalender**
Wichtige Branchenveranstaltungen

08 **Passiert auf makinggames.de**
Interessantes aus dem World Wide Web

10 **Making Games Tools**
Die Meinung der Profis zu wichtigen Updates

Rubriken

03 **Editorial**

78 **Was macht eigentlich ...** David Brevik?

80 **Ein Tag bei ...** Noumena Studios

82 **Vorschau**

82 **Impressum**

Titelstory Spielspaß

12 **Essay**
Spielspaß erklären
von Prof. Dr. Christoph Klimmt

14 **Case Study**
Spielspaß erforschen
von Jochen Peketz und Benedikt Grindel

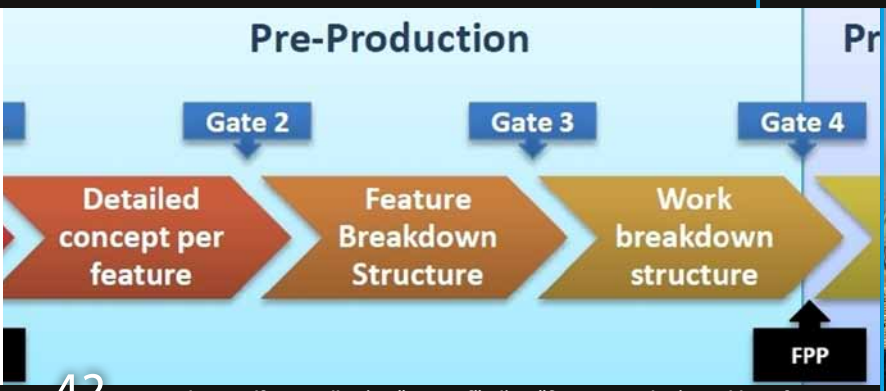
20 **Case Study**
Spielspaß analysieren
von Stephanie Kaiser

25 **Essay**
Spielspaß erfühlen
von Christian Schmidt

28 **Best Practice**
Spielspaß kalkulieren
von Daniel Dumont

14 Im Ubisoft GamesLab wird Spielspaß unter anderem mit Eyetracking erforscht.

46 Im Workshop – die automatische Generierung hunderter Waffenmodelle.



42 Der Producer Wolfgang Walk zeigt Lösungen für die größten Pre-Production-Fehler auf.

64 Das ambitionierte Studentenprojekt Gargoyles im Post Mortem.

Entwicklung

32 **Case Study**
Jagged Alliance Online
von Johannes Scharl

38 **Case Study**
Cross-Plattform bei TibiaME
von Benjamin Zuckerer

42 **Best Practice**
10 Pre-Production-Irrtümer
von Wolfgang Walk

46 **Workshop**
Generierung von Waffen-Modellen
von Jochen Winkler

Business

52 **Best Practice**
Internationales Community Management
von Szymon Wilkosz

56 **Interview**
So entstand Sims Social für Facebook
Im Gespräch mit Kristian Segerstrale
von Yassin Chakhchoukh

58 **Best Practice**
Tracking von Mobile Games
von Dennis Raspotnig und Rene Schölzel

61 **Recht (Serie)**
Entwicklungsverträge von Online-Spielen, Teil 3
von Claas Oehler

Campus

64 **Post Mortem**
Gargoyles – Stoned till Dawn
von Marvin Augenstein, Josef Vorbeck, Friedrich Brilling, Svenja Hildebrandt und Marko Plavsic

68 **Ausbildung**
Praktikumsbörse

Märkte

71 **Marktforschung**
GameStar- und GamePro-Leserdaten
von Yassin Chakhchoukh

72 **Firmenregister**
Die Branche im Überblick

Deutscher Entwicklerpreis 2011

In diesem Jahr ist alles anders – zumindest gilt dies für den Entwicklerpreis selbst, der nicht nur von Essen in die Rhein-Metropole Düsseldorf wandert, sondern mit 600 geladenen Gästen deutlich kleiner wird als in den vergangenen Jahren und damit zu seinen Wurzeln als »Familientreffen« der Branche zurückkehren möchte.

Gern würden wir Ihnen an dieser Stelle natürlich unsere persönlichen Eindrücke von der Preisverleihung aus dem Alten Kesselhaus schildern, doch leider mussten diese Zeilen bereits einige Tage vor dem Entwicklerpreis am 7. Dezember entstehen. Als Medienpartner wissen wir aber natürlich trotzdem über die Gewinner Bescheid, deshalb finden Sie auch eine vollständige Liste der Glücklichen im Kasten auf der rechten Seite.

Auch wenn die Location in diesem Jahr eine andere ist, gilt immer noch die gleiche Regel: Das Spiel mit dem höchsten Entwicklungs-Budget räumt die meisten Preise ab. Die fünf Awards für Crysis 2 (inklusive Bestes Deutsches Spiel) hat sich Crytek natürlich trotzdem mehr als redlich verdient. Die größte Überraschung ist in diesem Jahr sicherlich der Preis für das beste Game Design. Mit dem Sidescrolling-Shooter Zeit², ein ehemaliges Studentenprojekt, dürften nur die wenigsten gerechnet haben – am allerwenigsten die Entwickler von Brightside Games. Ohnehin war 2011 ein gutes Jahr für den Nachwuchs, wie die durchweg hochklassigen Nominierten für den Gameforge Newcomer Award eindrucksvoll beweisen. Platz 1 geht hier an die Black Pants Studios aus Kassel für ihr Physik-Adventure Tiny & Big.

Ebenfalls bemerkenswert ist die Tatsache, dass die Preise für den besten Publisher (Travian Games) und den besten Entwickler (Fishlabs) erstmals an Unternehmen aus dem Browser- und Mobile-Segment gehen.

Diskussionswürdig bleibt nach wie vor das Konzept der Publikumspreise. Denn das Online-Spiel Runes of Magic mag in vielerlei Hinsicht ein auszeichnungswürdiges Produkt sein, aber sicherlich nicht in der Kategorie »Bestes Casual Game«. Mal ganz abgesehen davon, dass ein Spiel, das größtenteils in Taiwan entwickelt wird, unserer Meinung nach gar nicht für den Deutschen Entwicklerpreis zugelassen werden dürfte.

Wir sind auf jeden Fall gespannt, wie die Branche den etwas exklusiveren Entwicklerpreis aufnehmen wird und gratulieren ganz herzlich allen Gewinnern.

Yassin Chakhchoukh

Die Sieger

Bestes Deutsches Spiel
Crysis 2 (Crytek)
Bestes Browsergame
Drakensang Online (Bigpoint Berlin)
Bestes Konsolenspiel
Crysis 2 (Crytek)
Bestes Mobile Game
The Great Jitters: Pudding Panic (kunst-stoff)
Bestes Adventure
Harveys Neue Augen (Daedalic Entertainment)
Bestes Jugendspiel
Harveys Neue Augen (Daedalic Entertainment)
Bestes Serious Game
Ludwig (Ovos media)
Bestes Sportspiel
Mein Fitness-Coach Club (Ubisoft Blue Byte)
Bestes Rollenspiel
Drakensang Online (Bigpoint Berlin)
Bestes Actionspiel
Crysis 2 (Crytek)
Bestes Lernspiel
Crazy Machines Elements (FAKT Software)
Bestes Strategiespiel
Patrizier IV – Aufstieg einer Dynastie
(Gaming Minds Studios / Kalypso Media)
Bestes Kinderspiel
Living Stories – Das verlorene Herz
(Daedalic Entertainment)
Beste Simulation
Der Landwirtschafts-Simulator 2011 (Giants Software)
Bestes Casual-Spiel
Runes of Magic – Lands of Despair
(Runewaker Entertainment / Frogster)
Bestes Handheld-Spiel
Demolition Dash (Chimera Entertainment / dreamfab)
Bestes Social Game
Tight Line Fishing (Socialspiel)
Bestes Game Design
Zeit² (Brightside Games)
Beste Story
Harveys Neue Augen (Daedalic Entertainment)
Beste Grafik
Crysis 2 (Crytek)
Bester Soundtrack
Crysis 2 (Crytek)
Bestes Art Design
Harveys Neue Augen (Daedalic Entertainment)
Beste Games-Technologie
CryEngine 3 (Crytek)
Gameforge Newcomer Award
Tiny & Big – Grandpa's Leftovers
(Black Pants Game Studio)
Bester Publisher
Travian Games
Bestes Studio
Fishlabs Entertainment

Global Game Jam: 2012 in sieben deutschen Städten

Über 1.500 Spiele, 6.500 Entwickler und 44 Länder – Ian Schreiber hat zum Global Game Jam 2011 beeindruckende Zahlen parat. Und auch im nächsten Jahr will der Director der weltweit größten kollaborativen Spieleentwicklung wieder ähnliche Erfolge vorweisen.

Daher finden sich vom 27. bis 29. Januar wieder rund um den Globus Spieleverrückte zusammen, die gemeinsam konzipieren, programmieren, modellieren. Deutschland ist mit sieben Standorten dabei: Leipzig, Bremen, Berlin, München, Hamburg, Magdeburg und Ludwigsburg.

Aus Platzgründen ist die Anmeldung für den Global Game Jam verpflichtend, außerdem wird eine Gebühr von etwa 20 Euro fällig, um die Kosten zu decken. Für die kulinarische Grundversorgung ist dann allerdings bereits gesorgt.

Am Ende wird jeder Teilnehmer sein Werk außerdem im Internet bewundern dürfen. Tolle Aktion: Die Berliner Jammer arbeiten in diesem Jahr mit dem Computerspielmuseum zusammen, wo alle Spiele während der Langen Nacht der Museen präsentiert werden.

www.globalgamejam.org

Gameforge baut rund 100 Stellen ab

Anfang November gab die Gameforge-Gruppe per Pressemeldung bekannt, dass man aufgrund einer Neuausrichtung der Firmenstrukturen rund 100 Mitarbeiter entlassen müsse. Die bisherigen operativen Einheiten Client-Games und Webgames wird es nicht mehr geben, stattdessen soll sich der Bereich Development in Zukunft auf die interne Entwicklung von Online Games konzentrieren, der Publishing-Bereich wiederum das Spieleportfolio sowie die Plattformstrategie verantworten. »Durch eine effizientere Organisationsstruktur können wir unseren Ressourceneinsatz optimieren und große Potenziale für das weitere Wachstum heben«, so Gameforge-CEO Alexander Rösner.

Als Folge der Umgestaltung werden bei Gameforge und der Tochter Frogster die Spiele Hellbreed sowie Mythos eingestellt, die nicht das erwartete Wachstum erbracht haben. Des Weiteren wird derzeit die Einstellung des von keen games entwickelten Star Trek – Infinite Space evaluiert, eines der aktuell größten Spieleprojekte von Gameforge. Allerdings wurden bereits Gespräche aufgenommen (unter anderem mit Bigpoint), die ein Co-Publishing und damit die Rettung des Projekts zum Ziel haben.

Der Wegfall bislang zentraler Marken sowie die Neuausrichtung der Unternehmensbereiche machen laut Mitteilung der Gameforge-Gruppe Kündigungen unausweichlich. »Wir bemühen uns, eine sozial verantwortungsvolle Lösung für die betroffenen Teammitglieder zu finden«, so Gameforge-CEO Alexander Rösner.

Drei Fragen an ...

Christoph Jennen

ist CFO der Gameforge AG.

Making Games Star Trek – Infinite Space von keen steht vor dem Aus, Hellbreed von Inflammables wurde bereits eingestellt. Wie sinnvoll ist es noch für deutsche Entwickler, mit euch bezüglich eines neuen Projekts in Kontakt zu treten?

Christoph Jennen Wir arbeiten gerne mit deutschen Entwicklern und bedauern es, dass einige Projekte nicht perfekt gelaufen sind oder ihnen der kommerzielle Erfolg verwehrt blieb. Es ist aber in unserem Geschäft Normalität geworden, dass nicht mehr jeder Launch auch ein Erfolg wird. Natürlich entwickeln wir weiter in Deutschland. Wir haben bei einem neuen, sehr hoffnungsvollen Spiel gerade die Pre-Production erfolgreich abgeschlossen. Hier arbeiten wir mit einem bekannten deutschsprachigen Entwickler zusammen. Dazu werden wir in Kürze mehr erzählen. Also: Wir suchen weiterhin Partner für Projekte und bleiben ein verlässlicher Auftraggeber. Wir glauben an Deutschland als Entwicklungsstandort und glauben auch an deutsche Entwickler.

Making Games Ist dies nun das Ende der bisherigen Wachstumsstrategie oder wollt ihr in absehbarer Zeit – wenn mit der Restrukturierung alles glatt läuft – wieder einstellen?

Christoph Jennen Unsere aktuelle Situation ist ein Zeichen dafür, dass die Gameforge sich vom Startup zu einem professionellen und reifen Business entwickelt. Nach Jahren des dreistelligen Wachstums (und der Akquisition einer Firma mit beinahe der halben Mitarbeiterzahl der Mutterfirma) war das ein normaler und notwendiger Schritt. Wir sehen das in vielen Wachstums-Industrien, in aller Regel gehen Unternehmen gestärkt aus solchen Prozessen hervor, insbesondere dann, wenn die Prozesse frühzeitig und konsequent eingeleitet werden. Für uns war dies ein logischer Schritt, um unsere erfolgreiche Unternehmensentwicklung weiter fortzuführen. Und natürlich werden wir in Zukunft auch wieder einstellen.

Making Games Was sind eure Schwerpunkte für die nun anstehende Konsolidierungsphase?

Christoph Jennen Wir betrachten die Konsolidierungsphase als nahezu abgeschlossen. Wir haben eine Reihe von harten, aber notwendigen Entscheidungen getroffen, seither richten wir den Blick nach vorn. Momentan konzentrieren wir uns auf unsere aktuelle Pipeline und unsere erfolgreichen Bestandsspiele, die durchaus noch Umsatzwachstum zeigen – ein gutes Beispiel ist Metin2, hier erwarten wir im Dezember den besten Umsatz-Monat seit Launch. Die Pipeline der Gameforge umfasst eine Reihe noch unangekündigter Titel, zumeist hochwertige Client-MMOs, und wird verstärkt durch das Frogster-Lineup. Das ist die beste MMO-Produktpipeline, welche die Gameforge-Gruppe je hatte.

Eventkalender

WWW makinggames.de
- Online-Eventkalender mit Suchfunktion und Veranstaltungsinfos

JANUAR 2012

Mobile Games Forum

Ort:	London, Hilton Tower Bridge
Termin:	25. bis 26. Januar 2012
Thema:	Konferenz
Kosten:	460 bis 1.500 Euro
Homepage:	www.mobilegamesforum.co.uk

Global Game Jam 2012

Ort:	7 Standorte in Deutschland
Termin:	27. bis 29. Januar 2012
Thema:	kollaborative Spieleentwicklung
Kosten:	Eintritt frei, Anmeldung erforderlich
Homepage:	www.globalgamejam.org

Making Games Talents München

Ort:	München, IDG Medienhaus
Termin:	28. Januar 2012
Thema:	Recruiting-Konferenz
Kosten:	29 Euro
Homepage:	www.makinggames.de/talents

FEBRUAR 2012

Casual Connect Europe

Ort:	Hamburg, Congress Center
Termin:	7. bis 9. Februar 2012
Thema:	Konferenz
Kosten:	350 bis 550 Euro
Homepage:	http://europe.casualconnect.org

MÄRZ 2012

GDC San Francisco

Ort:	Moscone Convention Center
Termin:	5. Februar bis 9. März 2012
Thema:	Konferenz
Kosten:	55 bis 1.100 Euro
Homepage:	www.gdconf.com

CeBIT

Ort:	Messe Hannover
Termin:	6. bis 10. März 2012
Thema:	Messe
Kosten:	19 bis 34 Euro
Homepage:	www.cebit.de

Hamburg Games Conference

Ort:	Bucerius Law School
Termin:	29. März 2012
Thema:	Konferenz
Kosten:	75 Euro
Homepage:	www.gamesconference.com

Gamecity Treff

Ort:	13. Stock
Termin:	29. März 2012
Thema:	Branchen-Treff
Kosten:	Eintritt frei, Anmeldung erforderlich
Homepage:	www.gamecity-hamburg.de

Veranstalten Sie mit!

Sie organisieren eine Podiumsdiskussion oder planen eine Party und möchten, dass die Branche davon erfährt? Dann schreiben Sie uns an **info@makinggames.de**, **Stichwort »Eventkalender«** und wir veröffentlichen Ihre Veranstaltung in unserer nächsten Ausgabe.

Passiert auf making games.de

App-Entwicklung direkt auf dem iPad

Mit Codify von Spieleentwickler Two Lives Left gibt es jetzt eine Entwicklungsumgebung für das iPad, mit der man ganz ohne weiteren Computer eigene Apps entwickeln kann. Als Sprache setzt die knapp 6 Euro teure App auf Lua, ergänzt wird das Scripting über ein rudimentäres Klassenkonzept. Spaßig sind die Anpassungen an die iPad-übliche Touch-Umgebung, um vor allem bei der virtuellen Tastatur die Nachteile gegenüber ausgewachsenen PC-IDEs auszugleichen. Dennoch fehlen mit Versionsmanagement und Debug-Tools einige wichtige Funktionen, zudem sollte für grundlegende Arbeiten trotzdem auf eine Bluetooth-Tastatur zurückgegriffen werden. Exportfunktionen gibt es ebenfalls nicht. Als wirkliche IDE mag Codify deshalb noch nicht ganz angekommen sein, aber die App ist ein interessantes Proof of Concept, wohin die Reise auch für Entwicklungsumgebungen gehen kann.

Sony trotz erhöhter PS3-Verkäufe im zweiten Quartal mit Verlust

Rund 27 Milliarden Yen Minus stehen nach dem zweiten Quartal des Fiskaljahres unter dem Strich bei Sony. Die umgerechnet über 251 Millionen Euro Verlust sind umso schmerzlicher, weil vor genau einem Jahr noch ein satter Gewinn von 295 Millionen Euro eingefahren wurde. Sony erklärte ihn über Probleme bei LCD-Displays, den Preisnachlass bei der PS3 und, wie es auch schon bei Nintendo der Fall war, über schlechte Wechselkurse. Betrachtet man nur die Playstation-Sparte, so verkündet der Geschäftsbericht auch positive Nachrichten. So konnte Sony sowohl bei der PS3 mit 3,7 Millionen verkauften Einheiten als auch bei der PSP mit 1,7 Millionen Einheiten 0,2 Millionen Einheiten mehr absetzen als im gleichen Quartal des Vorjahres. Insgesamt hat Sony damit 56 Millionen Einheiten weltweit verkauft, was bedeutet, dass sie wieder knapp an Microsoft mit 57,3 Millionen verkauften Xbox 360 herangerückt sind. Dagegen musste sich Sony mit weitaus weniger abgesetzten Stückzahlen bei der Software zufrieden geben: 8,1 Millionen Einheiten konnten verkauft werden, fast 2 Millionen Einheiten weniger als zuvor.

Unity wird es nicht für Windows Phone 7 geben

Unity CEO David Helgason erklärte in einem Interview mit Develop, warum man aktuell nicht Windows Phone 7 unterstütze und übte damit starken Druck in Richtung Microsoft aus. Windows Phone 7 sei ein relativ geschlossenes System, das verlange, dass man entweder in XNA oder Silverlight entwickle. Das könne Unity schon umsetzen und seine Engine kompatibel machen, es wäre aber zu viel Aufwand, als dass es sich lohnen würde. Deshalb schaue man vor allem in Richtung Windows Phone 8. Unity ist weit verbreitet bei Entwicklern, die ihre Apps für iOS und Android zugleich entwickeln. Aktuell ist das Windows Phone noch außen vor.

Adobe stellt mobiles Flash-Plugin ein

Adobe hat sich dazu entschlossen, sein Flash-Plugin für mobile Plattformen komplett einzustellen. Stattdessen wird mehr Wert auf das Adobe Integrated Runtime (AIR) und HTML5 gelegt. Damit geht Adobe den Weg, den Apple schon für sein iOS eingeschlagen hatte. Apple schloss Adobes Flash damals direkt aus, weil das Plugin zu langsam sei. Es kam zu einer langandauernden Schlammschlacht zwischen beiden Unternehmen. Von Adobes Entscheidung sind vor allem Android und RIMs Playbook OS betroffen. Der auch von Apple propagierte Standard HTML5 scheint damit weiter auf dem Erfolgsweg.

Nach Server-Crash wird ein ganzes MMO aufgrund fehlender Backups eingestellt

Es handelt sich zwar nicht um eine wirklich neue Meldung, aber es ist ein erstaunlicher Hinweis darauf, dass selbst große Firmen die Grundlage des wichtigsten Administrator-Mantras nicht beherzigen: Backup, Backup, Backup. Der japanische Developer Sankando und der Operator Hangame mussten in einer Pressemitteilung erklären, dass ihr MMORPG »M2« für immer verloren sei und daher eingestellt werde. Das Spiel musste aufgrund kritischer Fehler auf den Hauptservern heruntergefahren werden. Dabei gingen alle Daten verloren, Backups der Live-Server existierten nicht. Alle Spieldaten inklusive aller Spielerinfos sind verloren.

Studie: Südostasiatischer Online-Markt soll seinen Umsatz bis 2015 verdoppeln

Laut einer Studie von Niko Partners ist auch der Markt hinter den großen asiatischen Märkten China, Südkorea und Japan spannend für Online-Spiele-Publisher. So sei der südostasiatische Markt, der die Länder Indonesien, Malaysia, die Philippinen, Thailand und Vietnam umfasst, deutlich im Kommen und im Jahr 2015 vorrausichtlich eine Milliarde Dollar schwer. Für 2011 wird ein Umsatz von 474 Millionen Dollar erwartet, womit sich der Markt in den nächsten vier Jahren verdoppeln würde.

GameStop: »Kunden können sich nicht so viele Top-Titel leisten«

GameStop-Präsident Tony Bartel spricht derzeit von einigen Herausforderungen für sein Unternehmen. Der Grund: Im aktuellen Quartal seien zu viele Top-Titel erschienen, von denen sich die Kunden schlichtweg nicht alle leisten können. Zwar habe das Unternehmen den Umsatz um 2,5 Prozent auf 1,95 Milliarden Dollar im Vergleich zum Vorjahr erhöhen können, die Absätze seien aber hinter den Erwartungen zurückgeblieben. Wichtigste Titel waren in diesem Zeitraum Gears of War 3, Battlefield 3, Madden NFL 12, Batman: Arkham City und Dead Island. Für das Weihnachtsgeschäft erhofft sich Bartel bessere Absatzzahlen. Man setze große Hoffnungen in AAA-Spiele wie Modern Warfare 3, Skyrim, Assassin's Creed Revelations oder Star Wars: The Old Republic.

Branchen-Gezwitscher

@makinggames_de/interesting-people

@notch
In case you didn't realize, I'm on a post-release game playing vacation now. Recharging batteries and eating chips.
Markus Persson, Mojang

@carlmanneh
Ha. My 1-year old has rearranged all my apps so I can't find them. That makes me smile.
Carl Manneh, Managing Director von Mojang

@jeffgerstmann
Dancing on a table in Coruscant while my robot buddy watches. Let's call it a night, shall we?
Jeff Gerstmann, Gründer von GiantBomb

@georgeb3dr
I'm still playing Skyrim, but I put on pants to go to dinner. Will probably remove them soon. Also, Bethesda, MMO please.
George Broussard, 3D Realms

@TimOfLegend
Off to the courthouse to plead NOT GUILTY to DRIVING IN THE GODDAMNED BUS LANE. Even though I did it. But all the other lanes were closed!
Tim Schafer, Double Fine Productions

@jesseschell
Another day closer to a super fun announcement from Schell Games! Hint: it involves voting. And... moustaches.
Jesse Schell, Game Designer

@tinysubversions
I am at my ten year high school reunion, AND HAVING A GREAT TIME. I love these people. Is that weird? (Also very drunk.)
Darius Kazemi, Game Designer

@grumpygamer
I was cleaning my desk today and I found the Higgs boson those people are CERN are looking for. Where should I send it?
Ron Gilbert, Double Fine Productions

@ID_AA_Carmack
All the press hits about the Doom 3 source makes it difficult to google actual projects that have been started. Where are people working??
John Carmack, id Software

@gregcoomer
Writing this from a Win8 tablet. Feeling schizophrenic. The future looks bleak.
Greg Coomer, Valve

join the conversation

Making Games Links

Schon vor einigen Jahren hat die britische Autosendung Top Gear versucht, die berüchtigte Strecke Laguna Seca nur anhand von Gran Turismo 4 zu lernen und dann in Realität abzufahren. Das Ergebnis ist immer noch ein Klassiker.
www.makinggames.de/topgear

Der Australier Ian Andrews nennt sich selbst Videogame Photographer und hat einige sehenswerte Postkarten-Screenshots kreiiert.
www.makinggames.de/postcards

Wissenschaftler haben sich damit befasst, wer warum spielt und wie dies mit dem Zeitfokus der Person zusammenhängt – äußerst lesenswert.
www.makinggames.de/zeit

In einem Feature-Artikel für Gamasutra erklärt Josh Bycer das Design von Spielen mit einem konstant hohen Schwierigkeitsgrad wie etwa Demon's Souls. Ob der Ausdruck »Darwinian Difficulty« aber wirklich passend ist, mag und sollte gerne diskutiert werden.
www.makinggames.de/darwinian

Ja, es ist nicht das erste Spiel, das im Spiel Minecraft gebaut wurde, aber es ist wohl das vollständigste. Zudem ist der nachgebaute NES-Controller mehr als liebenswert..
www.makinggames.de/gameingame

Updates im Profi-Check
Making Games Tools

Keine guten Werkzeuge – keine guten Spiele! Experten aus der Branche sagen ihre Meinung zu aktuellen Programm-Updates sowie neuen Tools und verraten ihre persönlichen Software-Geheimtipps.

Honeytracks

https://honeytracks.com

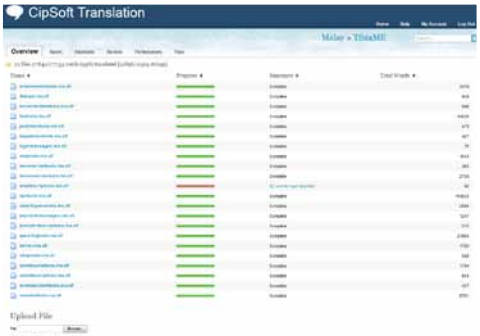
Sascha Ahlers
ist Community- und Marketing Manager bei SlipShift.

Honeytracks ist ein umfangreiches Tool zur Überwachung der Marketing- und Monetarisierung-Maßnahmen für Online-Games. Das Tool liefert auf den ersten Blick die KPIs zur Überwachung des Erfolgs, unterteilt in fünf Bereiche: User, Virality, Retention, Conversion und Revenue. Im Dashboard laufen die Statistiken zusammen und zeigen sofort – grafisch schön aufbereitet – die wichtigsten Daten wie MAU, DAU, Churn-Rate, User Lifetime, ARPU, ARPPU und so weiter. Die einzelnen Dashboards können dabei für jeden User-Account so angepasst werden, wie es einem beliebt. So kann der Geschäftsführer ohne lange Klickstrecken fast in Echtzeit die Zahlen kontrollieren. Der Marketing Manager wiederum kann tiefer gehen und sich genau ansehen, über welchen Kanal wie viele User kamen, wie diese konvertiert sind und welcher Revenue dabei herum kam. Speziell Performance-freudige Marketers werden ihre Freude daran haben, den RoI für einzelne Kanäle, Zielgruppen oder Länder zu berechnen. Das nächste Update fügt eine Historie für einzelne User hinzu, was etwa bei der Fraud-Detection helfen kann. Der Preis für das Tool richtet sich nach der Anzahl der eigenen User und skaliert damit zusammen mit dem Spiel – klasse!

Mit Honeytracks lassen sich **Marketing- und Monetarisierungs-Maßnahmen** überwachen. Der Preis des Tools richtet sich fairerweise nach der Anzahl der eigenen User.

Pootle

translate.sourceforge.net/wiki/pootle



Mihilfe des kostenlosen Tools Pootle lassen sich alle **Lokalisierungsarbeiten** übersichtlich und von überall aus auf einem Web-Server verfolgen.

Pootle ist ein kostenloses Web-Portal für Lokalisierungsarbeiten. Es steht unter der GNU General Public License (GPL) und kann kostenlos heruntergeladen und verwendet werden. Das Akronym Pootle steht für PO-based Online Translation / Localization Engine. Auch wenn die Abkürzung es nicht vermuten lässt, unterstützt Pootle das gängige XLIFF-Format vollständig. Viele große Projekte wie OpenOffice.org, Mozilla Firefox, Gnome oder KDE verwenden Pootle für ihre diversen Übersetzungen. Der Server ist schnell aufgesetzt und bietet eine benutzerfreundliche Oberfläche. Zusätzlich findet man im SourceForge-Projekt noch das Translate Toolkit – eine sehr umfangreiche und nützliche Sammlung von Python Tools. Diese bilden die technische Basis von Pootle, können aber auch für andere Konvertierungs- und Lokalisierungsarbeiten verwendet werden. Weil Pootle auf einem Web-Server läuft, hat man als Entwickler immer Zugriff auf den aktuellsten Stand der Übersetzung. Zusätzlich werden viele sinnvolle Checks für die Qualitätssicherung angeboten.

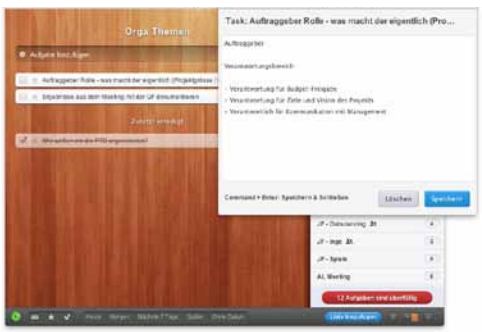
Benjamin Zuckerer
ist Produktmanager bei CipSoft.

Wunderlist

www.wunderlist.com

Oliver Feige
ist CTO von Travian Games.

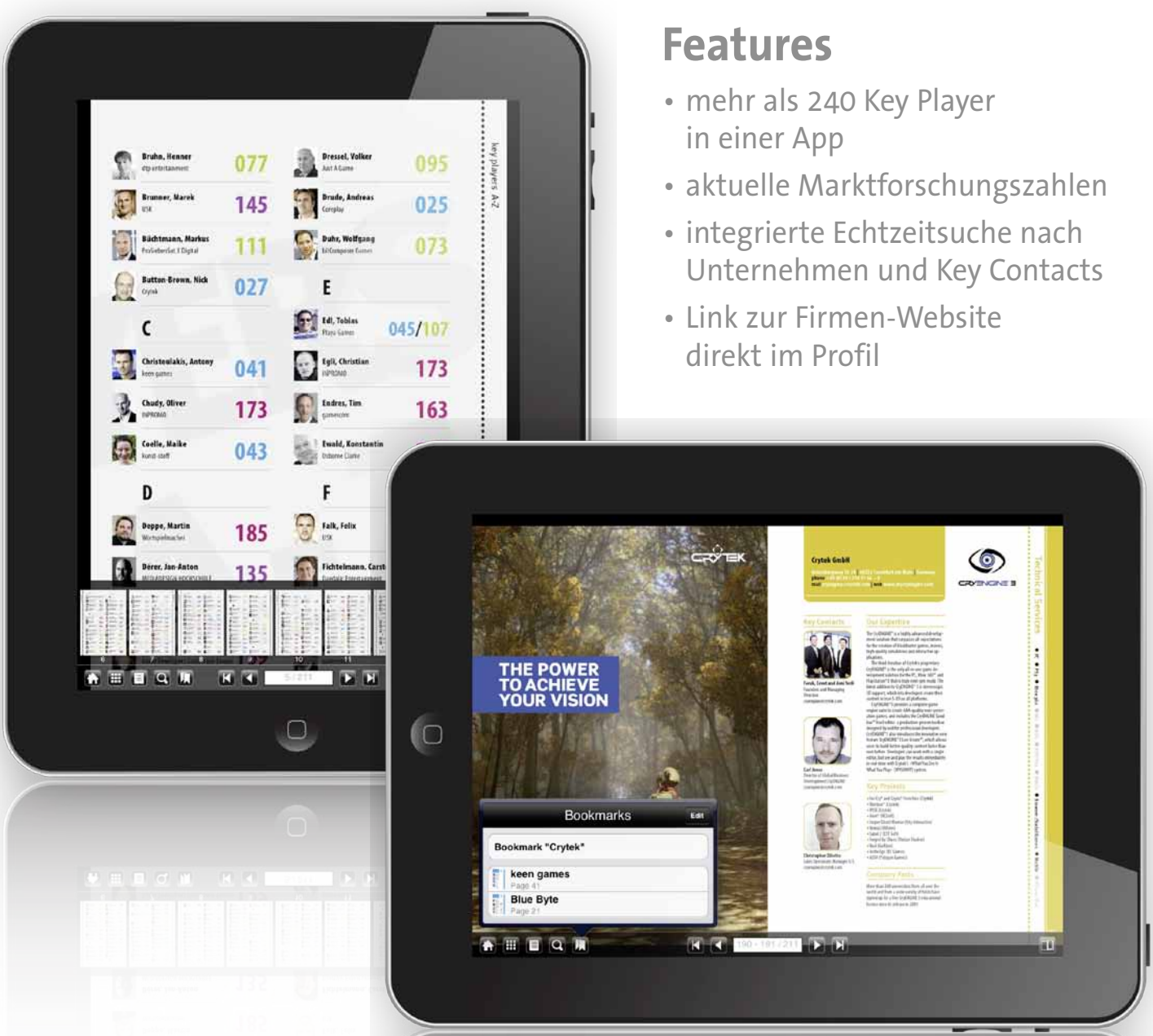
Wunderlist ist ein plattformübergreifender Taskmanager. Im Vordergrund steht eine simple und intuitive Bedienung. Aufgaben können einfach erstellt, mit Fälligkeitsdaten sowie Notizen versehen und durch Drag & Drop schnell sortiert werden. Erinnerungen sorgen dafür, dass keine Aufgabe unter den Tisch fällt. Zwei Features machen Wunderlist jedoch zu einem besonders praktischen Tool. Die Möglichkeit Listen für bestimmte Aufgabenfelder anzulegen und diese mit Kollegen auch via Cloud teilen zu können. Über diese Listen können ganze Aufgabengebiete unkompliziert und vollständig an ein ganzes Team übergeben werden. Die Cloud-Synchronisierung der Aufgaben macht die Nutzung auf unterschiedlichen Endgeräten äußerst praktikabel. Durch die Nutzung des Amazon Web Services im Backend sind darüber hinaus Sicherheit und Datenverfügbarkeit permanent gewährleistet.



Mit Wunderlist lassen sich **Taks und ganze Aufgabengebiete** plattformübergreifend über die Cloud an alle Teammitglieder verteilen.

Das Kompendium der Deutschen Games-Branche – jetzt auch auf dem iPad!

KEY PLAYERS APP



Features

- mehr als 240 Key Player in einer App
- aktuelle Marktforschungszahlen
- integrierte Echtzeitsuche nach Unternehmen und Key Contacts
- Link zur Firmen-Website direkt im Profil

DOWNLOAD NOW

www.makinggames.de/keyplayers

Available on the App Store

Der Stand der Wissenschaft
Spielspaß erklären

Computer- und Videospiele sind längst auch ein Thema an Hochschulen und Universitäten. Professor Christoph Klimmt liefert Einblicke, wie die Forschung den Spielspaß untersucht und welche Erkenntnisse sie bereits gesammelt hat.

Prof. Dr. Christoph Klimmt

ist Professor am IJK Hannover.

Der Autor ist Professor am Institut für Journalistik und Kommunikationsforschung (IJK) der Hochschule für Musik, Theater und Medien Hannover. Spielspaß und andere Themen rund um Computerspiele gehören seit gut zehn Jahren zu seinen Hauptforschungsgebieten.
christoph.klimmt@ijk.hmtm-hannover.de

Wissenschaftliche Erklärungen sind Antworten auf die Frage »Warum?«. Diese Definition ist nur scheinbar banal, denn über sie gelangt man zügig zu den großen erkenntnistheoretischen Fragen der Wissenschaft. Anstelle eines Crashkurses in Wissenschaftsphilosophie seien hier aber nur die zwei wesentlichen Ansatzpunkte erläutert, mit denen die Sozialwissenschaften Phänomene wie etwa »Spielspaß« zu erklären versuchen.

1. Theoretische Erklärungen folgen dem Prinzip der Deduktion. Dabei werden Ereignisse, etwa die Freude, die ein Gamer bei »Call of Duty: Modern Warfare« erlebt, zurückgeführt auf allgemeine Gesetzmäßigkeiten. Der Spielspaß, den CoD entfacht, wird hier also erklärt als Manifestation allgemeiner menschlicher Motivation: Junge Männer, so könnte eine theoretische Erklärung lauten, haben entwicklungsbedingt Freude daran, ihre heldenhafte Männlichkeit zur Schau zu stellen. Genau das ermöglicht ein Gefecht am virtuellen Hindukusch, ergo lässt sich der Spielspaß von CoD auf das allgemeine Prinzip der Freude an demonstrierter Männlichkeit zurückführen. Klingt seltsam? Nun ja, das Beispiel soll ja nur ein Prinzip verdeutlichen. Aber eigentlich ist diese Erklärung nach all dem, was die Wissenschaft über Spielspaß weiß, ganz gut.

Mithilfe von empirischen Untersuchungen haben Wissenschaftler inzwischen belegt, dass Frauen an **wettbewerbsorientierten Spielen** wie Call of Duty weniger Spaß haben als Männer.

2. Empirische Erklärungen zielen darauf ab, Unterschiede im Auftreten eines Phänomens (zum Beispiel Unterschiede im »CoD-Spielspaß« zwischen Männern und Frauen) auf Unterschiede in anderen Merkmalen zurückzuführen. Dass nicht jeder Spieler und jede Spielerin beim selben Spiel den gleichen Spaß erlebt, ist ein alter Hut. Empirische Erklärungen versuchen nun, solche Unterschiede zu messen und systematisch mit Unterschieden in anderen Merkmalen zu verbinden. So haben der Amsterdamer Kollege Tilo Hartmann und ich vor ein paar Jahren etwa per Umfrage ermittelt, dass Frauen im Durchschnitt weniger Interesse an wettbewerbsorientierten Computerspielen haben als Männer. Daher finden wir eine empirische Erklärung für den Geschlechterunterschied beim »CoD-Spielspaß«: Frauen bereitet ein wettbewerbsintensives Spiel wie Call of Duty weniger Freude als Männern, die im Vergleich geradezu nach Wettbewerb lechzen.

Spielspaß ist Spannung

So viel zur Frage, was Wissenschaft mit »Spielspaß erklären« eigentlich meint. Wie erklären wir nun also Spielspaß? Die Beantwortung dieser Frage steht zunächst einmal vor einem Problem der großen Vielfalt. Nicht nur Menschen unterscheiden sich in ihrem Spielspaß (siehe das Männer/Frauen-Beispiel vorhin). Die gleichen Menschen erleben in unterschiedlichen Phasen der Beschäftigung mit dem gleichen Spiel höchst unterschiedliche Quantitäten und Qualitäten von Spielspaß – es gibt also auch Variation zwischen Spielsituationen. Und nicht zuletzt gibt es natürlich auch erhebliche Variation im Spielspaß, die auf die Vielfalt der Spielformen, -genres und -modi zurückzuführen ist. Trotzdem finden sich natürlich einige »Essentials«, die bei vielen Spielern, in vielen Spielsituationen und über verschiedene Spielformen hinweg relevant sind. Aus meiner Sicht ist Spannung die mit Abstand bedeutsamste Facette von Spielspaß. Wie bei einem Krimi entsteht die Erfahrungsqualität Spannung typischerweise dadurch, dass die Spieler mit Ereignissen konfrontiert werden, die ihnen nahegehen und deren Ausgang vorerst ungewiss bleibt. Spannend wurde es für mich bei Call of Duty beispielsweise, als gleich in der ersten Mission ein defekter

Panzer verteidigt werden musste: Der Spieler (ich) wollte die Aufgabe lösen, die Gegner besiegen, die virtuellen Kameraden retten (Punkt 1: Die Ereignisse waren dem Spieler – mir – wichtig). Zugleich hielten die zahllosen Gegner eine taktisch überlegene Stellung und setzten dem eigenen Team massiv zu. Ob die Situation gemeistert werden konnte, war also ungewiss (Punkt 2 der Formel für Spannung). Das Zusammentreffen von emotionaler Eigenbeteiligung und von Ungewissheit erzeugt den Zustand unangenehmer physischer Erregung und höchster Aufmerksamkeit, den wir gemeinhin als Spannung beschreiben und den viele Leser, Zuschauer und Gamer als ausgesprochen »entertaining« erleben. In Games, egal ob Call of Duty, World of Warcraft oder Starcraft, reihen sich viele Momente solcher Spannung aneinander. Daher erklärt sie einen Großteil des Spielvergnügens über Spieler, Situationen und (viele) Genres hinweg. Spannung speist sich einerseits aus Aspekten der Story (wie geht es weiter mit den sympathischen Figuren, die mir ans Herz gewachsen sind?). Andererseits – und dieser Fall dürfte beim Gaming wesentlich wichtiger sein – entsteht Spannung aus Leistungsdruck: Spieler müssen und wollen eine Herausforderung meistern, aber ob es ihnen gelingt, steht jedes Mal für eine Weile in den Sternen. Genau dafür sorgt die KI oder der menschliche Gegenspieler.

Spielspaß ist Psychologie

Andere wissenschaftliche Erklärungen für Spielspaß sind ähnlich universell wie Spannung, teilweise jedoch auch begrenzt auf bestimmte Spielertypen oder Spielsituationen. Die Freude am Erfolg, der Stolz auf die eigene Leistung ist eine Mechanik des Spielspaßes, die in vielen Momenten im Vordergrund steht. Ganz grundsätzlich schätzen Gamer die Erfahrung eigener Wirksamkeit und Einflussnahme, also das Erleben direkter Veränderungen in der Spielwelt als Folge ihrer Entscheidungen und Mausklicks. Bei Multiplayer Games gehören Erlebnisse von sozialer Anerkennung und die Glücksmomente des perfekten gemeinsamen Timings zu den typischen Erscheinungsformen des Spielspaßes. Die Liste dieser Beispiele ließe sich noch sehr lange fortsetzen – es steckt jede Menge und jede Menge unterschiedlicher Spaß in Spielen.

Um solche mehr oder weniger universellen Mechanismen des Spielspaßes zu erkennen, greift die akademische Forschung auf unterschiedliche Theorien, Modelle und Konzepte zurück. Dazu gehören vor allem Ansätze aus der Motivationspsychologie. Die Selbstdeterminationstheorie von Richard Ryan und Edward Deci ist ein prominentes Beispiel: Sie fasst menschliche Motivation in drei Bedürfnisse, nämlich Autonomie, Kompetenz und soziale Zugehörigkeit. Die Parallelen zu Games – insbesondere Multiplayer-Spielen – lassen sich leicht erkennen. Auch Grundlagen aus der Entwicklungspsychologie ziehen wir zur Modellierung von Spielspaß heran. Das schon erwähnte Konzept

Raid in World of Warcraft: Theorien zu Gruppenprozessen, Gruppenidentifikation sowie soziologische Grundlagen bieten in diesem Bereich wissenschaftliche Anknüpfungspunkte.

der Selbstwirksamkeit stammt von dort, ebenso die Erkenntnisse über Maskulinität als bestimmendes Thema in der männlichen Jugendphase, die nutzbar gemacht werden, um den Spielspaß von Shootern zu erklären. Schließlich bieten sozialpsychologische Grundlagen wie die Theorien zu Gruppenprozessen und Gruppenidentifikation genauso wie soziologische Grundlagen (etwa zur Jugendkultur) Anknüpfungspunkte, gerade um neuere Facetten des Spielspaßes zu erklären, die mit Online-Games verbunden sind. Vor allem das Miteinander in Online-Spielwelten hat die Vielfalt der Erfahrungsqualitäten beim Gaming dramatisch erweitert.

Spielspaß ist mehr als Datenauswertung

Die meiste Forschung zum Spielspaß kommt interessanterweise ohne einen Austausch mit Game-Designern zustande. Dass Entwickler Spielspaß erforschen, analysieren und sogar berechnen können, wie in diesem Heft nachzulesen ist, haben wir Wissenschaftler natürlich schon gehört. Das macht die Wissenschaft einerseits langsam im Nachvollziehen aktueller Entwicklungen. Über Facebook-Games wissen wir bislang beispielsweise sehr wenig. Andererseits schafft Wissenschaft Erklärungen für Spielspaß, die dabei helfen, Games in die Medienlandschaft und den gesellschaftlichen Kontext einzuordnen. Die Besonderheiten von Interaktivität bei der Genese von Spaß zu rekonstruieren, gelingt eben mithilfe von (psychologischen) Theorien weitaus besser als allein anhand von Serverdaten eines Online-Spiels. Insofern können und sollten Game Design und Spielspaßforschung einen engeren Austausch pflegen: Die Wissenschaft kann gewiss wichtige Impulse aus der Praxis erhalten. Und vielleicht finden Entwickler ja die ein oder andere wissenschaftliche Erklärung für Spielspaß auch ein wenig inspirierend.

Prof. Dr. Christoph Klimmt

Literaturempfehlungen

- Peter Vorderer & Jennings Bryant (Hrsg.). (2006). **Playing Video Games: Motives, Responses, Consequences**. Mahwah: Lawrence Erlbaum Associates.
- Thorsten Quandt, Jeffrey Wimmer & Jens Wolling (Hrsg.). (2009). **Die Computerspieler. Studien zur Nutzung von Computergames**. Wiesbaden: VS Verlag für Sozialwissenschaften.

Das Blue Byte GamesLab
Spielspaß erforschen

Die Jungs von Blue Byte erläutern, wie sie mithilfe eines eigens eingerichteten Labors und systematisch durchgeführten Tests ihre Spiele auf Zugänglichkeit, Lernkurve und Langzeitmotivation optimieren.

Jochen Peketz

ist GamesLab Manager bei Ubisoft Blue Byte.

Jochen Peketz startete 1999 bei Piranha Bytes in die Games-Branche. Hier skriptete er die KI für Gothic. Nach einem kurzen Ausflug als Redakteur bei einer Konsolenspielzeitschrift und Dozent für Computerkurse für Kinder stieg er bei Phenomic (später EA Phenomic) als Level Designer für Spellforce ein. Im Jahr 2008 wurde Jochen Lead Level Designer für Siedler 7 bei Blue Byte, um dann im Jahr 2010 Manager des GamesLab zu werden und dieses aufzubauen.

Benedikt Grindel

ist Head of Live Operations bei Ubisoft Blue Byte.

Benedikt Grindel ist seit 1998 in der Branche. Der diplomierte Mathematiker hat als Group Marketing Director für Ubisoft Spiele vermarktet und als Business Development Manager für Ludiwap / Gameloft die ersten Geschäftskontakte zu deutschen Mobilnetzbetreibern hergestellt, bevor er 2001 als Producer bei Blue Byte direkt in die Entwicklung eingestiegen ist, wo er unter anderem an Siedler 5, Siedler 6 und Siedler 7 mitgearbeitet hat. Seit einem Jahr verantwortet er bei Blue Byte als Head of Live Operations den Betrieb der Online-Spiele.

Wer kann sich nicht erinnern: Gerade fast 100 Deutsche Mark für ein Spiel hingeblättert, die zweistündige Installationsprozedur (inkl. Neuinstallation des Betriebssystems und Updates sämtlicher denkbaren Treiber) hinter sich gebracht, das Spiel gestartet und dann auf den Spielspaß gewartet. Aber vor dem Spaß kam die Arbeit: Die Steuerung war schwierig, die ersten Aufgaben mühsam, die 250-seitige Anleitung nicht wirklich hilfreich. Macht nichts: Irgendwann würde man es finden, das geniale Spiel, es würde sich offenbaren, wenn man nur so weit kommt. Und das stimmte oft genug! Viele PC-Titel der 90er-Jahre strotzten nur so vor Spieltiefe, Langzeitmotivation und tollen Ideen, aber wirklich einsteigerfreundlich waren sie nur selten.

Seitdem hat sich viel geändert. Vor allem Titel aus dem Hause Nintendo zeigten, wie motivierend und toll Spiele von der ersten Sekunde an sein können, während man bei so manchem PC-Produkt noch den Anleitungstext für die Verwendung der kontextsensitiven rechten Maustaste studierte.

Heute ist die Zugänglichkeit von Spielen auf jeder Plattform ein Muss. »Accessibility« heißt das neudeutsch, und wenn ein Spiel heute nicht »accessible« genug ist, bedeutet das ein großes Risiko für den wirtschaftlichen Erfolg.

Warum ein GamesLab?

Für die Entwicklung von Spielen ist »Accessibility« eine große Herausforderung. Das liegt in der Natur der Sache: Wenn man einige Jahre an einem Produkt arbeitet, gewöhnt man sich an ein schlechtes Interface oder an ein wenig einsteigerfreundliches Tutorial. Um wirklich beurteilen zu können, wie gut die Startphase eines Spiels funktioniert, benötigt man immer wieder neue Spieler, die es ausprobieren. Und man muss sehr gut auswerten, was sie verstehen, wie lange sie für die einzelnen Schritte benötigen und wie sich das über mehrere Versionen entwickelt – und hoffentlich verbessert.

Das ist eine sehr anspruchsvolle Aufgabe, für die Ubisoft vor einiger Zeit das GamesLab ins Leben gerufen hat. Diese Spezialabteilung gab es zunächst nur in der Zentrale, inzwischen gibt es GamesLabs in fast allen Studios der Ubisoft-Gruppe. Vor knapp zwei Jahren – mit den Erfahrungen aus Produktionen wie Die Siedler 7 und ANNO 1404 – startete das GamesLab bei Blue Byte die schwierige Mission, all seine Spieler von der ersten Sekunde an glücklich zu machen.

Mit dem GamesLab sollen die Zugänglichkeit der Spiele verbessert und die Lernkurve analysiert werden. Im Fokus sind in Düsseldorf natürlich erst einmal die Produkte von Blue

Spielen unter Beobachtung: Dank regelmäßiger Playtests im **Blue Byte GamesLab** verbessern die Entwickler systematisch Zugänglichkeit und Lernkurve ihrer Spiele.

Byte. Es werden aber auch regelmäßig Tests für andere Ubisoft-Projekte durchgeführt.

Die Kandidaten

Im Bereich der Marktforschung ist der »Mann-von-der-Straße-Test« ein geflügelter Begriff. Trotzdem laufen wir natürlich nicht vor jedem Test erst mal durch die Gegend, um Kandidaten zu rekrutieren. Und doch gibt dieser alte Name eine unserer Zielsetzungen gut wieder: Wir arbeiten mit »typischen Endkunden« zusammen.

Wie kommt man an solche Endkunden? Schließlich ist es wichtig, dass wir immer wieder neue Probanden haben. Hat jemand einmal den Einstieg in ein Spiel erlebt und an einer Testrunde bei uns teilgenommen, ist er für weitere Untersuchungen zum gleichen Thema sozusagen verbrannt. Das bedeutet: Für bestimmte Titel laden wir insgesamt weit über 100 Leute zu uns ein, um iterativ immer wieder Ergebnisse zu erhalten und mit geänderten Spielversionen erneut zu untersuchen, ob sich das Spiel in die richtige Richtung entwickelt.

Die Auswahl der Endkunden richtet sich nach den teilweise sehr unterschiedlichen Zielsetzungen. Ein Beispiel: Es ergibt natürlich keinen Sinn, den neuesten Teil von ANNO auf Einsteigerfreundlichkeit zu testen und dazu nur langjährige Serienfans einzuladen. Auf diese Weise erhält man keine validen Ergebnisse. Um die Auswahl unserer Probanden zu erleichtern, pflegen wir eine Datenbank, in der Informationen zu jedem Spieler (Spielverhalten, Vorlieben, Hardware etc.) hinterlegt sind. So können wir mit Filtern, die wir über die Datenbank laufen lassen, schnell die gewünschten Profile heraussuchen und die Testpersonen einladen.

Um die Datenbank aktuell zu halten und mit immer neuen Kandidaten zu bestücken, werden die verschiedensten Kanäle genutzt. Der klassische Anlaufpunkt ist natürlich unsere Website (www.bluebyte.de/GamesLab), aber auch mit Aufrufen über unsere Facebook-Gruppen, in den Spieleforen oder per Pressemitteilung (Letzteres gerade für Online-Medien) haben wir sehr gute Erfahrungen gemacht.

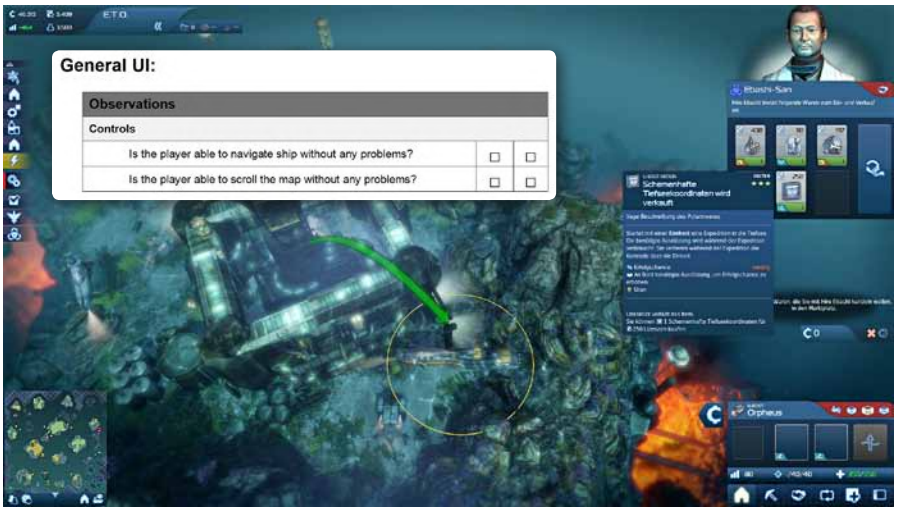
Die Testbogen

Die Tests verfolgen grundsätzlich eine bestimmte Zielsetzung, die im Vorfeld von den Entwicklungsteams definiert wird. Vielleicht sollen nur bestimmte Interface-Neuerungen oder der Ersteinstieg ins Spiel oder das Verständnis bestimmter Spielelemente getestet werden.

Der erste Schritt im GamesLab ist es dann, die Testversion genauestens im Hinblick auf die Thematik zu untersuchen: Die Version wird gespielt, analysiert, Probleme werden identifiziert.

Mit diesen Resultaten geht es anschließend an die Erstellung unserer Testbogen. Damit steht und fällt der Erfolg der Analysen; deshalb muss alles so wissenschaftlich und objektiv wie möglich sein. Häufig hören wir in diesem Moment das Argument, dass Spaß doch nicht messbar sei – es ginge doch um Spiele! Diese Aussage

Über eine eigens eingerichtete **Website** sucht Blue Byte permanent nach Testpersonen für das GamesLab.


General UI:
Observations
Controls
Is the player able to navigate ship without any problems?
Is the player able to scroll the map without any problems?

Mit diesen **Fragen** aus dem Testbogen überprüften die Entwickler, wie intuitiv die Schiffs- und Karten-Navigation in ANNO 2070 funktionierte und welches Feintuning noch vonnöten war.

stimmt aber nur teilweise. Es gibt auch bei Spielen objektiv messbare Maßstäbe, die eingehalten werden sollten, wenn der Nutzer nicht verärgert oder frustriert werden soll. Das kann die Zeit sein, die ein gewisser Vorgang benötigt, oder die Anzahl der Mausklicks, die gerade noch tolerabel ist, um eine Aufgabe zu lösen. Auch Textlänge und Komplexität der Darstellung spielen eine Rolle (siehe auch **Eyetracking-Kasten**).

Recruiting

Sobald der Testbogen erstellt und mit dem Entwicklerteam abgestimmt ist, geht es ans Rekrutieren der Teilnehmer. Die Datenbank wird auf geeignete Kandidaten hin durchsucht, diese werden anschließend eingeladen. Idealerweise machen wir das per Telefon, aus zwei Gründen: Einerseits kann man gewisse Daten am Telefon leichter und schneller abgleichen, andererseits sind die Reaktionszeiten offenbar wesentlich kürzer als zum Beispiel per E-Mail. Sind alle Kandidaten gefunden, erhalten sie von uns eine Bestätigung (diesmal schriftlich per Mail) mit al-

Für einen reibungslosen Testablauf ist es unerlässlich, dass die Probanden vollständig präparierte **Arbeitsplätze** vorfinden und sofort loslegen können.

len wichtigen Daten wie Termin, Dauer, Anfahrtsbeschreibung etc.

Pro Aufgabenstellung und Testerprofil benötigen wir mindestens acht Probanden, um valide Ergebnisse zu erzielen. Benötigen wir Resultate zu mehreren Profilen – etwa für Casual Gamer und Hardcore Gamer – müssen wir somit zweimal acht Personen einladen. Die Definition der für einen Test notwendigen Zielgruppen ist daher besonders wichtig. Wir versuchen uns dabei auf wenige, klar definierte Zielgruppen zu konzentrieren. Es besteht sonst die Gefahr, sich schnell zu verzetteln. Die Zahl »acht« ist hierbei einerseits ein Erfahrungswert, andererseits hat zum Beispiel die FH Düsseldorf bei einer wissenschaftlichen Eyetracking-Auswertung (siehe Kasten auf Seite 19) für ANNO 2070 ebenso diese Anzahl Probanden verwendet.

In ANNO 2070 lassen sich **Häuserreihen** ziehen, was von den Testern jedoch kaum eingesetzt wurde. Nur durch Interviews war herauszubekommen, dass der Grund dafür nicht im User Interface lag.

Testablauf

Am Testtag finden sich die Probanden bei uns im Studio ein. Nach dem Erledigen der Formalitäten (Aushändigen eines Besucherausweises, Unterzeichnen der Vertraulichkeitsvereinbarung etc.) werden die Spieler in unser Testlabor geführt. Dieser speziell eingerichtete Raum ist mit jeder erdenklichen Hardware ausgestattet. Bei Testbeginn ist bereits alles vorbereitet (Spiele installiert, Accounts angelegt, Papier und Stifte für Notizen liegen bereit) und es kann direkt losgehen. Das klingt nach einer Kleinigkeit, bedeutet aber viel Vorbereitungszeit und sollte bei der Planung keinesfalls unterschätzt werden. Die Probanden nehmen an den Geräten Platz und bekommen eine kurze Einweisung in den Test. Es wird zum Beispiel auf Bugs der Testversion hingewiesen oder es wird den Spielern erklärt, wo der Fokus des Tests liegt.

Ab diesem Zeitpunkt sind die Kandidaten auf sich allein gestellt. Je zwei Tester werden von einem Testleiter beobachtet, und die Beobachtungen werden in die Testbogen eingetragen. Primär geht es dabei um die vorher ausgearbeiteten Problemstellungen, aber auch andere Auffälligkeiten werden notiert, damit sie später für die Auswertung zur Verfügung stehen. Die Spieler sind dazu angehalten, alle Probleme, auf die sie stoßen, allein zu lösen. Nur so können wir sicherstellen, dass wir valide Ergebnisse erhalten, etwa wenn ein Tester auch nach einigen Minuten die gestellte Aufgabe noch nicht angehen kann, weil er die Bedienelemente nicht versteht.

Trotzdem haben die Tester natürlich die Möglichkeit, um Hilfe zu bitten, wenn sie gar nicht weiterkommen. Wer eine Stunde vor dem Spiel sitzt und nicht weiterkommt, bringt zu wenig verwertbare Ergebnisse. In diesen Fällen notieren wir die Problemstelle und zeigen dann, wie bzw. wo es weitergeht.

Tests dieser Art dauern in der Regel zwei bis vier Stunden. Die in dieser Zeit gesammelten Daten sind noch gut auswertbar und die Konzentration der Spieler bleibt auf einem guten Niveau. Um die Zeiten einhalten zu können, arbeiten wir teilweise auch mit Speicherständen.

Sonderfall Lernkurve

Einen etwas anderen Fokus haben Tests zum Thema Lernkurve. Typischerweise dauern diese Tests länger, teilweise durchaus mehrere Tage. Hier geht es darum, die Kampagne eines Spiels am Stück durchspielen zu lassen, um zu sehen, wo Schwierigkeiten auftreten. Auf Probleme kann zum Beispiel hinweisen, wenn ein Level besonders lange dauert oder wenn der Spieler eine Aufgabe zu oft probieren muss.

Häufig trifft der Spieler an dieser Stelle auf eine neue Spielmechanik, die nicht verständlich genug eingeführt wird. Eine Lösung wäre hier, das Level Design so anzupassen, dass der Spieler die neuen Elemente zunächst spielerisch üben kann.

Eine zweite Problematik lässt sich wesentlich direkter adressieren: der Schwierigkeitsgrad einzelner Spielabschnitte. Wenn viele Spieler immer an derselben Stelle scheitern, kann das häufig schon durch einfache Änderungen am Balancing gelöst werden.

Das abschließende Interview

Während der Testrunden sammeln wir viele Daten, die im anschließenden Interview abgeglichen und teilweise vertieft werden. Ein einfaches Beispiel für unser Produkt »Die Siedler Online« wäre zu prüfen, ob der Spieler in einem bestimmten Zeitraum das erste Gebäude gesetzt hat. Im Interview kann man ihn fragen, ob er den Gebäudebau als schwierig empfunden hat.

Um im Interview gute Ergebnisse zu erhalten, benötigt man viel Fingerspitzengefühl – denn jeder Kandidat tickt anders. Mit der Zeit sammelt man Erfahrung, welche Spieler man einfach reden lässt, bei wem man noch mal nachhaken kann (oder muss), um tiefergehende Erkenntnisse zu gewinnen. Die Bandbreite der Personen, die bei uns für Tests anreisen, reicht von spielbegeisterten Familien über den klassischen Hardcore-Zocker bis hin zu Facebook-Spielern.

Ein ganz wichtiger Punkt bei den Befragungen ist, diese immer nur mit einer Person gleich-

zeitig durchzuführen. Führt man die Interviews in einer Gruppe, wird man immer wieder das Problem bekommen, dass mindestens ein Spieler (bewusst oder unbewusst) die Meinungsführerschaft übernimmt und so die anderen Tester beeinflusst. Befragt man jeden Probanden einzeln, umgeht man dieses Problem, weil sich schüchterne oder vorsichtige Charaktere nicht innerhalb einer Gruppe behaupten müssen, sondern vom Interviewer behutsam an die Thematik herangeführt werden können.

Bei den Interviews werden die Beobachtungen mit den Erlebnissen der Spieler abgeglichen. Es lassen sich zwar viele Erkenntnisse durch reines Beobachten gewinnen, aber bei bestimmten Aspekten ist es wichtig, diese noch durch ein Interview zu vertiefen. Ein Beispiel: Bei ANNO 2070 war zwar zu beobachten, dass Spieler den alternativen Häuserbau (Ziehen statt einzelnem Setzen der Wohnhäuser) nicht verwenden, aber der Grund dafür war nicht erkennbar. Es gab manche Spieler, die das Feature zwar verstanden hatten, aber durch alte Teile der Serie darauf trainiert waren, Häuser einzeln zu setzen, und das auch mochten. An dieser Stelle war nur durch ein Interview herauszubekommen, dass es eine bewusste Entscheidung der Probanden war und sie damit zufrieden waren. So wurde klar, dass das Feature gut verstanden wurde und daher keine weitere Arbeit investiert werden musste, um es zu verbessern.

Die Testauswertung

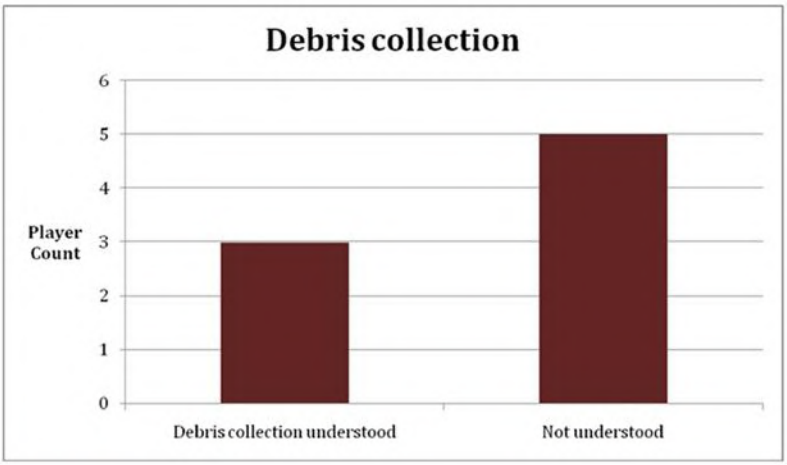
Sind alle Tests abgeschlossen und alle Daten erhoben, folgt die Auswertung. Die Reporte werden so erstellt, dass jeder Leser versteht, was passiert ist und wo die Probleme liegen, selbst wenn er nicht persönlich bei den Tests anwesend war. Jeder Bericht beschreibt genau das Test-Setup und die jeweilige Gruppe der Probanden, um ein gutes Gefühl zu vermitteln, wie und was getestet wurde. Anschließend beginnt der Testbericht mit einer sogenannten »Executive Summary«, die die wichtigsten Ergebnisse kurz und knapp zusammenfasst.

Hauptteil des Reports ist eine Liste aller aufgetretenen Probleme. Hier werden die Beobachtungen festgehalten, warum Probleme aufgetreten sind und wie sie angegangen werden können. Das GamesLab ordnet auch die Wichtigkeit der Probleme ein und gibt dem Team so Hinweise, welche Dinge als Erstes angegangen werden sollten.

Im Anschluss an den Versand des Berichts erfolgt natürlich noch eine Nachbesprechung. Hier wird diskutiert, was in Folge des Tests mit welcher Priorität angepasst wird und wann der nächste Test angesetzt werden soll, um diese Änderungen zu überprüfen. Und ein weiterer Test-Zyklus nimmt seinen Lauf ...

Wissenschaft und Training

In den inzwischen fast zwei Jahren, in denen wir hier in Düsseldorf ein eigenes GamesLab haben, konnten wir viele Probleme identifizieren und an die Teams weitergeben. Durch das Feedback der Spieler wurden viele Features deutlich und nachweisbar verbessert. Das reicht von optimierten Minen in Die Siedler Online bis hin zur verbesserten Nutzerführung oder einem besseren Balancing in ANNO 2070.



DESIGN BREAK — **Context:** We asked the player if he know how the debris collector works.
Observation: 5 out of 8 players did not understand that debris is a resource.
Problem: Debris is understood as an endless resource.
Cause: Miss managed economies that appear later in the game to be more or less stuck.
Suggestion You should introduce debris as a resource properly.

Die **Testauswertung** macht nicht nur auf Probleme, mögliche Gründe und Konsequenzen aufmerksam, sondern hebt auch Verbesserungspotenziale hervor.

Es ist wichtig, den Markt zu beobachten und auf dem Laufenden zu bleiben, welche wissenschaftlichen Entwicklungen für uns relevant sind. Daher arbeiten wir intensiv mit Universitäten, Fachhochschulen und privaten Hochschulen wie der MDH zusammen. So ermöglichen wir Studenten, in Zusammenarbeit mit dem GamesLab ihre Bachelor- und Masterarbeiten zu verfassen. Damit haben wir sehr gute Erfahrungen gemacht.

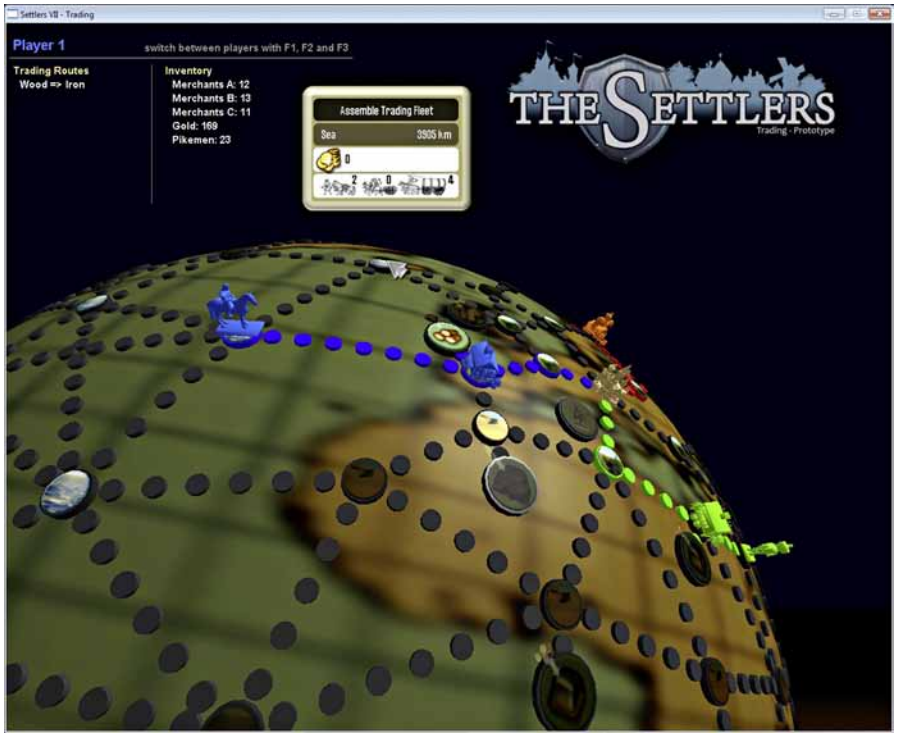
Die Arbeit im GamesLab ist spannend, weil wir immer mit neuen Leuten und an neuen Projekten arbeiten. Wir begleiten Projekte zwar auch über einen langen Zeitraum, aber wir begleiten immer mehrere Projekte gleichzeitig. Dabei müssen wir sehr flexibel sein: Wird ein Fitnessspiel wie »My Fitness Coach Club« getestet, muss das GamesLab-Team sich halt auch mal in diese Materie einarbeiten und ein paar Trainingsstunden bei einer professionellen Trainerin nehmen. Man sieht: Der Job ist nicht nur abwechslungsreich, man bleibt beim richtigen Produktmix auch körperlich fit!

Jochen Peketz

Riskante Gameplay-Neuerungen wie das Siegpunkte-System von **Die Siedler 7** werden bei Blue Byte von Expertentestern immer wieder auf ihre Funktionalität und das Balancing überprüft, um etwaige Probleme so früh wie möglich im Entwicklungsprozess aufzuspüren.

Testen von Langzeitmotivation

Die Qualität eines Spiels hängt von vielen Faktoren ab. An erster Stelle steht die »Accessibility« – ein Spiel muss leicht zugänglich sein, die Bedienung und die Spielregeln müssen Schritt für Schritt erklärt werden, und das auf eine jederzeit motivierende Weise. Das GamesLab setzt hier an, immer wieder testen neue Spieler die ersten Schritte im Spiel. So wird systematisch dafür gesorgt, dass die Zielgruppe des Spiels gut in das Spiel hineinfindet; die Pflicht ist erfüllt. Was nun folgt, ist die Kür: Die Spieler sollen lange dabeibleiben. Dafür benötigt das Spiel vor allem Tiefe und eine hervorragende Balance.



Ein bereits spielbarer **Prototyp des Handels** von Die Siedler 7. Expertentester fanden heraus, dass er im Vergleich zu Kampf und Wissenschaft zu schwach war.

Während der Entwicklung von Die Siedler 7 war das eine besondere Herausforderung. Wir hatten uns dazu entschlossen, eine neue Spielmechanik einzuführen: die Siegpunkte. Mit Siegpunkten sollte das Spiel dynamischer und Skirmish- sowie Mehrspielerpartien spannend bis zur letzten Sekunde werden. Vom Design waren wir überzeugt, denn mehrere Prototypen hatten gezeigt, dass das System grundsätzlich funktioniert. Die Frage war nun: Wie können wir diese neue Spielmechanik optimal testen und tunen, bis sie wirklich rund ist? Wenn die Balance nicht stimmt oder wenn es gar einen Königsweg zum Sieg geben sollte, wäre das Spiel nicht mehr spannend, sondern das genaue Gegenteil.

Das ganze Team spielt!

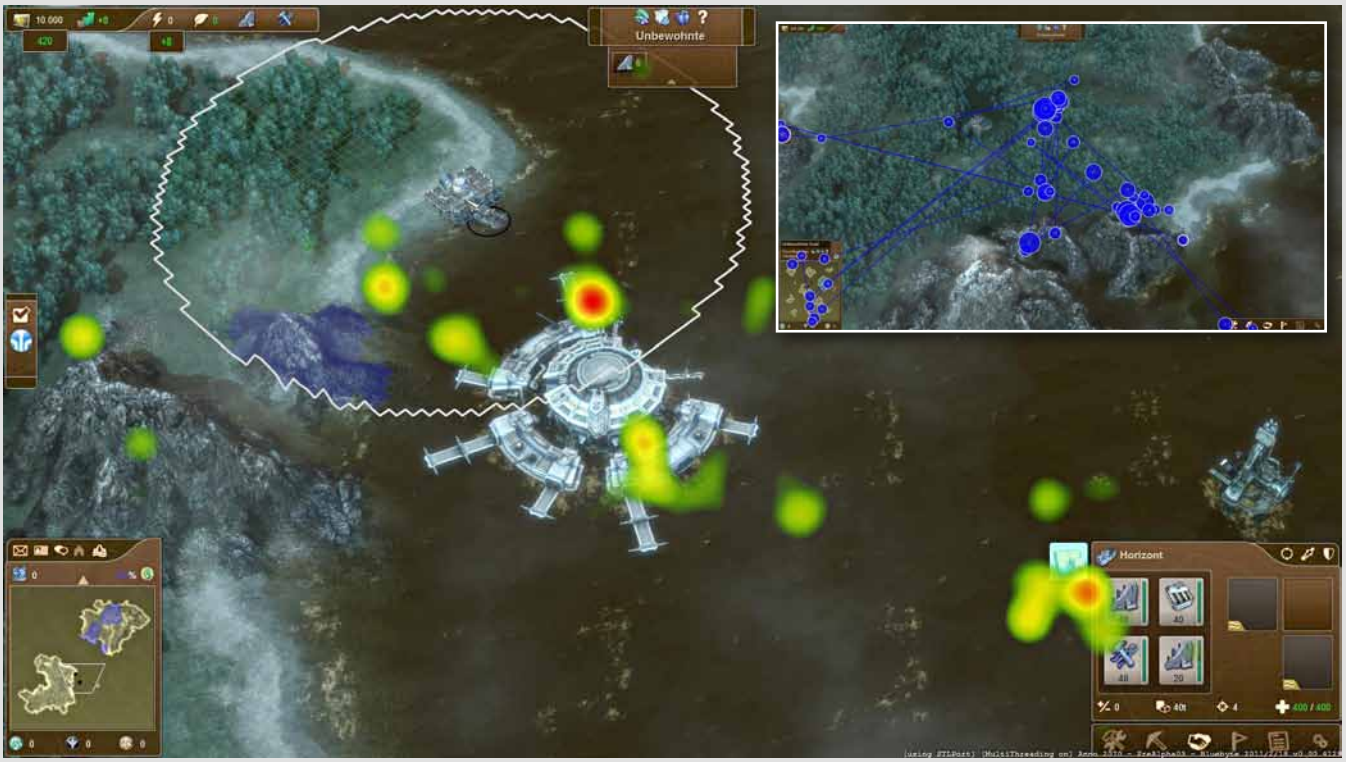
Einen Teil der Antwort lieferte uns Bruce Shelley, der während der Entwicklung von Die Siedler 7 als Consultant bei Blue Byte gearbeitet hatte. Bruce war seit seinen Anfängen in der Spieleindustrie, als er mit Sid Meier an Titeln wie Railroad Tycoon und Civilization arbeitete, von »Design by Playing« überzeugt: So früh wie möglich wurde in seinen Projekten eine einfache spielbare Version erstellt, gespielt und dann immer weiterverbessert. Bei Siedler 7 setzten wir das um, es wurde ein regelmäßiger »Play Day« eingeführt, an dem das ganze Team spielte. Danach wurde anonym ein Fragebogen ausgefüllt, der uns viele gute Hinweise gab. Neben den Play Days, die einmal im Monat stattfanden, luden wir jeden Entwickler zweimal im Monat zu einer zusätzlichen Play Session ein und beobachteten ihn dabei. So bekamen wir bereits in der frühen Entwicklungsphase einen guten Eindruck, ob und wie das Spiel funktioniert. Und auch das Team hatte immer einen guten Überblick, wie weit die Entwicklung war – ein nicht zu unterschätzender Nebeneffekt.

Die Expertentester

Noch wichtiger für das Balancing des Spiels war allerdings die Gruppe der »Expertentester« – ein besserer Name ist uns nicht eingefallen. Diese Gruppe von vier Personen, die regelmäßig mit Bruce verstärkt wurde, hatte die Aufgabe, das Spiel perfekt zu spielen. Wir verboten ihnen sogar, Bugs einzutragen oder sonstige typische Tester-Aufgaben zu übernehmen. Sie mussten (!) den ganzen Tag spielen.

Gerade zu Beginn war das schwierig. Als die Expertentester die Arbeit aufnahmen, lief das Spiel noch nicht stabil und es war nicht immer möglich, eine Partie durchzuspielen. Nicht zuletzt durch den Druck der Tester besserte sich das, und nun ging es darum, alle denkbaren Strategien zu probieren und zu analysieren. Weil das Siegpunkte-System sehr dynamisch ist und Siedler 7 außerdem drei verschiedene Ausbreitungsarten anbietet (durch Kampf in der Spielwelt, durch Erforschung auf der Technologie-Karte und durch das Besetzen von Handelsposten auf der Weltkarte), war das eine anspruchsvolle und

Experimente mit Eyetracking



Heat Maps kumulieren die Eyetracking-Daten von ganzen Probandengruppen (das ANNO-Interface war zum Testzeitpunkt noch Work in Progress).

Bei ANNO 2070 haben wir zum ersten Mal mit der Eyetracking-Technologie gearbeitet. In Zusammenarbeit mit der Fachhochschule Düsseldorf hat die Studentin Anika Marschollek bei Blue Byte ihre Bachelorarbeit zum Thema »Usability-Analyse von Computerspielen am Beispiel von Ubisoft ANNO« geschrieben. Sie wurde von Prof. Dr. Sven Pagel und Prof. Dr. Markus Dahm wissenschaftlich sowie von Alexander Jürgens technisch betreut. Auf Seiten von Blue Byte wurde das Projekt von Jochen Peketz unterstützt. Dabei hat sich Eyetracking als spannender Ansatz erwiesen, Usability-Analysen auf einen anderen Level zu heben. Es gibt mehrere Möglichkeiten, Daten auszuwerten:

1. Die erste und offensichtlichste Methode ist es, einfach Videos des Blickverlaufs und des Spielgeschehens zu »grabben«. So kann man sehr gut sehen, an welcher Stelle des Screens der Spieler versucht, Informationen zu finden, die ihm helfen, das aktuelle Problem zu lösen. Ergibt sich bei einer genügend großen Anzahl von Probanden das Bild, dass an einer anderen Stelle gesucht wird als an der Stelle, wo die Informationen hinterlegt sind, sollte das UI überarbeitet werden.

2. Die alternative Methode besteht darin, dass man anhand des Blickverlaufs sehr gut feststellen kann, ob Texte wahrgenommen und gelesen werden. Es gibt Schwellenwerte für die Fixation, die überschritten werden müssen, um mit großer Sicherheit sagen zu können, dass ein Text gelesen wurde. Auf diese Art kann man sogar ermitteln, ob häufig an einer bestimmten Stelle der Lesevorgang abgebrochen wird. Ist dies der Fall, sollte die Länge einer Textpassage noch einmal überarbeitet werden.

Blickverlaufdaten erfordern viel Handarbeit in der Auswertung. Es gibt bisher noch keine guten Software-Lösungen, die helfen, die Daten automatisiert zu erfassen. An dieser Stelle kommen sogenannte »Heat Maps« ins Spiel. Heat Maps kumulieren die Fixationsdaten von ganzen Probandengruppen und erleichtern so die Auswertung. Auch hier kann man wertvolle Informationen erlangen, um Interface-Verbesserungen vorzunehmen. Eyetracking bietet somit in der richtigen Versuchsumgebung die Möglichkeit, einen Anfangsverdacht zu erhärten und mit objektiven Methoden zu beweisen.

langwierige Aufgabe. Die Expertentester mussten alle Varianten beherrschen. Und sie mussten dabei immer wieder neue Dinge lernen. So stellte sich zum Beispiel heraus, dass die Handelsoption zu schwach war: Wenn ein Spieler auf Handel setzte, hatte er gegen einen guten Gegner keine Chance zu gewinnen. Nach der entsprechenden Anpassung war der Händler dann zu stark, und es ging wieder von vorn los.

Experten-Ergebnisse

Unsere Expertengruppe fand durch das ständige Spielen während der Testphase selbst kleinste Ungereimtheiten, die uns sonst entgangen wären. Sie trugen entscheidend zum Optimieren des Spiels bei, indem sie die Game und Level Designer ständig mit Feedback versorgten. Die letztendliche Verantwortung der Umsetzung blieb dabei beim Design Team.

Wenn ein Spiel im Wesentlichen auf bekannten Spielmechaniken aufbaut, ist eine gute Qualität mit weniger Aufwand sicherzustellen. Wenn man aber an entscheidender Stelle Neues ausprobiert, ist die Einrichtung einer Gruppe spezialisierter Gameplay-Tester eine sehr gute Maßnahme. Die hervorragenden Spielbewertungen (im Schnitt 87%) und die bis heute sehr hohen Spielerzahlen (bis zu 10.000 täglich aktive Spieler) zeigen das eindrucksvoll.

Benedikt Grindel

Monster World – eine Fallstudie

Spielspaß analysieren

Stephanie Kaiser

ist Product Lead bei wooga.

Stephanies erster Berufswunsch war es, Postbote zu werden. Denn was gibt es Schöneres, als Briefe zu erhalten? Ihre Begeisterung für Brücken ließ sie kurz über eine Karriere als Brückenbauer nachdenken. Während sie dieses Ziel heimlich noch nicht aufgegeben hat, ist sie in der Zwischenzeit Product Lead bei wooga geworden, wo sie heute Social Games entwickelt. wooga ist der drittgrößte Hersteller von Social Games weltweit. Als Product Lead mit 15 Mitarbeitern entwickelte Stephanie mithilfe von strikt metrikorientierten Designmethoden und wöchentlichen Entwicklungszyklen den Facebook-Titel Monster World. Monster World ist mit heute 1,8 Millionen täglich aktiven Nutzern das größte Monsterspiel auf Facebook. Derzeit leitet sie neben der Monsterpflege die Entwicklung von drei neuen Spielen. Vor ihrer Zeit bei wooga hat Stephanie als Product Manager bei Jamba und MTV Networks gearbeitet. Zuvor studierte sie an der Humboldt Universität Französische Philologie und Informationswissenschaft. Roboter und Ninjas sind ihre heimliche Leidenschaft.

Abbildung 1: Die **Wachstumskurve** für die ersten Monate nach der Veröffentlichung im April 2010.

Im April 2010 wurde Monster World auf Facebook veröffentlicht. Seitdem wird das Spiel von wooga wöchentlich aktualisiert. Product Lead Stephanie Kaiser erklärt im Detail, warum Metriken die Grundlage für jede einzelne Designentscheidung bilden.

Nicht alle Spiele sind sofort nach ihrer Veröffentlichung ein Hit. So war es auch bei Monster World, einem Farming-Spiel, das im April 2010 auf Facebook an den Start ging. Heute ist Monster World mit rund 1,8 Millionen täglich aktiven Spielern (DAU = Daily Active User) woogas erfolgreichster Titel. Und noch immer wächst das Spiel stetig.

Vor diesem Hintergrund werde ich in diesem Artikel einige Fallbeispiele aus dem Entwicklungsprozess beschreiben, strukturiert anhand der folgenden Themen: Engagement, Viralität und Monetarisierung.

Engagement

Monster World war nach dem Launch im April 2010 kein sofortiger Erfolg. Wir haben das Spiel daraufhin Schritt für Schritt im Laufe der Monate verbessern können. Wenn man sich die Wachstumskure nach der Veröffentlichung anschaut, kann man sehr schön unseren Arbeitszyklus erkennen (**Abbildung 1**). Jeden Dienstag veröffentlichen wir eine neue Version. Und jeden Dienstag lässt sich an der Wachstumskurve eine weitere Steigerung erkennen.

Wir haben zunächst damit begonnen, Funktionalitäten im Bereich des Spieler-Engagements zu verbessern. Sowohl virales Wachstum als auch gute Monetarisierung sind nutzlos, so lange die Spieler nicht zu einem Spiel zurückkehren. Die wichtigsten Kennzahlen im Bereich von Spieler-Engagement sind die Rückkehrquoten nach 1, 3 und 7 Tagen sowie der »Sticky Factor« (monatlich aktive Spieler geteilt durch täglich aktive Spieler). Abgesehen davon haben wir jeden einzelnen Teilschritt des Tutorials analysiert. Wie viele Spieler brechen zwischen den einzelnen Schritten weg, wie viele Spieler

Jede einzelne Designentscheidung für **Monster World** wird bei wooga durch Metriken untermauert, um die Spieler möglichst lange zu binden – wie etwa auch Stephanies Mutter, deren Garten hier zu sehen ist.

schließen das Tutorial komplett ab? Mithilfe von A/B-Tests haben wir einzelne Schritte des Tutorials verbessert, um möglichst vielen Spielern den Einstieg in das Spiel zu erleichtern.

In einem A/B-Test leiten wir einen gewissen Prozentsatz unserer Spieler in eine Version von Monster World, während eine Kontrollgruppe eine andere Version spielt, die das zu testende Feature nicht enthält. Im Anschluss analysieren wir die relevanten Kennzahlen und entscheiden uns für die Weiterentwicklung einer der beiden Versionen. Diese Tests müssen simultan stattfinden, um den Einfluss externer Faktoren (Wetter, Feiertage etc.) auszuschließen.
Wir haben beispielsweise eine Version des Tutorials getestet, in der die Spieler gezwungen wurden, exakt die Aktion durchzuführen, die unser Tutorial-Charakter Mr. Tentacle vorschlug. Die Kontrollgruppe hingegen konnte völlig frei entscheiden was sie als Nächstes tun wollte. Mr. Tentacle gab lediglich Hinweise.

Das Ergebnis war überraschend. Wir hatten angenommen, der Spieler würde die Freiheit bevorzugen. Als wir uns dann die Ergebnisse ansahen, mussten wir akzeptieren, dass die Spieler geführt werden wollen. Deutlich mehr Spieler haben das Ende des Tutorials erreicht, wurden ihnen die dazu notwendigen Aktionen gezeigt. Heute sehen alle Spieler diese Version des Tutorials.

Generell integrieren wir keine Funktionalitäten in das Spiel, nur weil ein Produktmanager denkt, sie seien gut. Sie schaffen es nur dann ins Spiel, wenn sie durch Zahlen belegt sind. Ein sehr gutes Beispiel ist die »Monster-Wahl«, die zu Beginn der Einstieg in das Spiel war (**Abbildung 3**). Denke ich daran zurück, gefällt mir die Auswahl des Monsters noch immer. Nach der Zahlenanalyse mussten wir jedoch feststellen, dass wir zu viele Spieler in diesem Schritt verloren. Überraschenderweise waren die Spieler nicht stärker motiviert, zu dem Spiel zurückzukommen, wenn sie am Anfang ihr eigenes Monster auswählen konnten. Also haben wir das Feature gestrichen.

Aufgrund der großen Anzahl von Spielern (ca. 300.000 DAU nach 2 Monaten) konnten A/B-Tests in vergleichbar kurzer Zeit statistisch relevante Ergebnisse zeigen. A/B-Tests waren von Anfang an eines unserer wichtigsten Optimierungsinstrumente.

Durch das Streichen irrelevanter Features und die Verbesserung einzelner Schritte im Tutorial konnten wir die Rückkehrrate neuer Spieler erhöhen. In der Folge erweiterten wir das Spiel um ein Missionssystem (**Abbildung 4**), das den Spielern Aufgaben in ihrer Monster World gab. Diese Aufgaben erhöhen das Engagement der Spieler, auch längerfristig im Spiel zu bleiben.

Nachdem wir Woche für Woche am Tutorial und an den Missionen gearbeitet hatten, begann das Spiel langsam und stetig zu wachsen. Die Rückkehrrate der Spieler war deutlich höher als bei der Erstveröffentlichung von Monster World und wir konnten dazu übergehen, die Viralität des Spieles zu verbessern.

Abbildung 2: Screenshot eines beispielhaften **A/B-Tests des Tutorials**. Links die Lightbox-Version, in der die Spieler eine bestimmte Aktion ausführen müssen. Rechts ohne Lightbox und mit freier Aktionswahl.

Abbildung 3: Der Auswahlbildschirm für die Monsterfamilie ist ein Beispiel für eine Funktionalität, die **nach Auswertung eines A/B-Tests gestrichen** wurde.

Viralität

Die viralen Kanäle auf Facebook haben sich in den letzten Jahren stark verändert. Auf den ersten Blick könnte man annehmen, diese Änderungen seien schlecht für Entwickler. Wenn man sich die Änderungen aber im Detail ansieht, wird man feststellen, dass sie aus Spieler-Perspektive sehr positiv sind. Und weil es sowohl Facebooks als auch woogas Ziel ist, zufriedene Spieler zu haben, sind die aktuellen Änderungen auch für uns Entwickler sinnvoll. Noch vor zwei Jahren konnten Spiele eine unbegrenzte Anzahl von Feedposts senden. So wurden sie von Nutzern schnell nur noch als Spam wahrgenommen. Heute ist es viel wichtiger, ein gutes Spiel zu entwickeln, als die Ausnutzung viraler Kanäle zu perfektionieren.

Feedposts werden heute nur noch den Nutzern gezeigt, die ein Spiel auch spielen. Diese Änderung erhöht die Relevanz des Facebook-Streams für den jeweiligen Nutzer. Relevanz ist der wichtigste Filter in einem zu jeder Zeit überfüllten Nachrichtenstrom. Mit der vergleichsweise kleinen Änderung haben sich Feedposts

Abbildung 4: Das **Missionssystem** wurde erst nachträglich in Monster World integriert. Es erhöhte das Engagement der Spieler deutlich, die Spieler kehrten häufiger zur Erledigung ihrer Aufgaben zurück.

Abbildung 5: Der **Händler Robert** fragt immer wieder nach ganz bestimmten Pflanzen. Die Spieler können ihre Freunde um Hilfe bitten, um den Deal schneller abzuschließen.

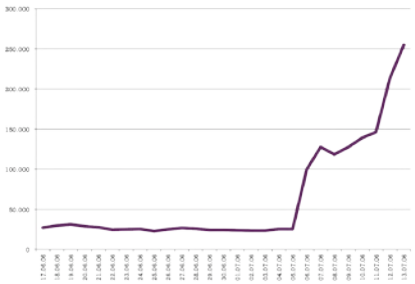


Abbildung 6: Jeder Feedpost im Nachrichten-Stream muss bei wooga **relevant** sein. Der Empfänger einer Nachricht erhält immer ein Geschenk. Nach der Einführung des Geschenks ist die Klickrate stark angewachsen.

von einem Viralitäts- zu einem Retention-Feature gewandelt. In der Folge hat Facebook die Anzeige von Spieleranfragen verändert. Die Anfragen (sogenannte Requests) werden gesammelt und als rote Ziffer über dem Globus-Icon in der Menüleiste angezeigt. Anfragen können an die komplette Freundesliste eines Spielers gesendet werden, auch an Nutzer, die das Spiel noch nicht gespielt haben. Mit dieser neuen Funktionalität sind die Anfragen mittlerweile zu einem wichtigen viralen Kanal geworden.

Relevanz

Jede virale Nachricht sollte für den Spieler von hoher Relevanz sein. Sowohl für den Sender als auch für den Empfänger müssen Interaktionen im Spiel eine wertvolle Funktion erfüllen. Verschafft ein Feedpost dem Spieler einen Vorteil im Spiel, wird er diesen Feedpost wahrscheinlicher mit seinem Freundeskreis teilen.

Ein gutes Beispiel aus Monster World ist der Robert-Feedpost. Robert ist der Roboter im Spiel, der immer einen Deal parat hat und dem Spieler Pflanzen abkauft. (Abbildung 5). Er zahlt gut. Der Spieler hat die Möglichkeit, seine Freunde um Hilfe für den Deal mit Robert zu fragen. Jeder Freund, der auf den Feedpost klickt, sendet Pflanzen an den jeweiligen Spieler, der mithilfe seiner Freunde so schneller im Spiel vorankommt.

Im Gegenzug sollte der Empfänger des Feedposts ebenfalls einen Vorteil im Spiel haben, wenn er diesen klickt. So bekommt jeder Spieler, der seinen Freunden Pflanzen sendet, ein Geschenk: Münzen oder Erfahrungspunkte. Als wir die zusätzlichen Belohnungen veröffentlicht haben, hat sich die Antwortrate auf unsere Feedposts drastisch erhöht. (Abbildung 6).

Zusätzlich haben wir weitere Features implementiert, die die Viralität ebenfalls positiv beeinflussen sollten. Ein sehr erfolgreiches Beispiel ist die Einführung des Monster Babys (Abbildung 7). Die Spieler beschäftigen sich umso mehr mit einem Spiel, je stärker die emotionale Bindung zu den Charakteren ist. Das traurige Monster Baby spricht das Helfersyndrom im Spieler an.

Das Baby weint, bis der Spieler das Spielzeug gebaut hat, das es gern haben möchte. Für den Bastelsatz wird eine bestimmte Anzahl unterschiedlicher Materialien benötigt. Diese kann der Spieler entweder beim Ernten von Pflanzen finden (mit relativ niedriger Wahrscheinlichkeit), durch die Hilfe von Freunden (via Feedpost) erhalten oder einfach kaufen. Das Baby spricht die Emotionen des Spielers an. Die Anzahl der Feedposts hat sich rapide erhöht, nachdem wir das Baby eingebaut hatten. Niemand konnte es ertragen, es weinen zu sehen.

Nach der Fertigstellung des ersten Spielzeugs zieht das Baby in den Garten des Spielers ein und lebt dort zwischen den Pflanzen. Ab jetzt wird es täglich nach Essen und Zuneigung fragen. Wenn seine Bedürfnisse nicht erfüllt werden, beginnt es wieder zu weinen. Für die tägliche Pflege erhalten Spieler einen Bonus.

Vor einiger Zeit hatten wir einen Besucher zu einem Usability Test bei uns im wooga-Büro. Er war ein begeisterter Spieler von Monster World und erzählte uns, das Spiel sei mittlerweile fester Bestandteil seines Lebens. Er hatte von Gerüchten gehört, das Baby könne sterben, wenn es nicht täglich gefüttert und gestreichelt würde. Ausprobieren kann er es nicht, er möchte das Baby ja behalten. Aber soviel sei gesagt: Unsere Monster World ist eine ganz freundliche Welt.

Als Facebook seine viralen Kanäle änderte, sind wir von der Nutzung der Feedposts zum Verschicken von Anfragen gewechselt, um das vollständige virale Potenzial von Facebook ausnutzen zu können. Anfragen können an alle Freunde verschickt werden, nicht nur an dieje-

Abbildung 7: Charaktere wie das traurige **Monster Baby** binden die Spieler stärker an das Spiel. Das kleine Monster weint so lange, bis der Spieler ihm mit Hilfe von diversen Materialien ein Spielzeug gebaut hat.

nigen, die das Spiel bereits nutzen. Auf diese Weise werden auch neue Spieler auf Monster World aufmerksam gemacht.

Monetarisierung

Nach der Optimierung von Monster World in den Bereichen Engagement und Viralität konnten wir uns näher mit dem Thema Monetarisierung befassen. Indem wir die Anzahl der Käufer und ihre tatsächlichen Einkäufe analysierten, haben wir das Offensichtliche festgestellt: Verbrauchsgüter monetarisieren erheblich besser als rein dekorative Elemente. Sie erzeugen einen konstanten Kaufzyklus, weil sie ständig im Tausch gegen Spielvorteile verbraucht werden.

Eines dieser Verbrauchsgüter in Monster World ist der Zauberstab. Zauberstäbe können im Spiel entweder dazu genutzt werden, von Helfermonstern angefressene Pflanzen wieder zu reparieren oder das Wachstum einer Pflanze zu beschleunigen. wooga hat sich mittlerweile als weltweit größter Anbieter von virtuellen Zauberstäben etabliert.

Als wir die Bedeutung von Verbrauchsgütern erkannt hatten, führten wir ein neues ein: woogoo (Regenbogenschleim). Anfang des Jahres wurde eine Reihe von Features implementiert, die mit woogoo zu tun haben. Robert hat jetzt eine Freundin: Roberta. Roberta möchte ähnlich wie Robert ständig etwas vom Spieler kaufen. Statt nach Pflanzen fragt Roberta nach Produkten, für deren Herstellung woogoo benötigt wird. Die Spieler erhalten woogoo, indem sie Pflanzen ernten, an Robert verkaufen, ihre Freunde um Hilfe bitten oder es kaufen (**Abbildung 8**).

Wir haben den woogoo-Vorrat im Spiel so abgestimmt, dass er grundsätzlich nicht ausreichend ist. Trotzdem haben die Spieler die Chance, ohne Bezahlung an den begehrten Rohstoff zu gelangen. Am Ende reicht es dennoch nicht aus.

Um alle Wünsche Robertas zu erfüllen oder den Erntehelfer benutzen zu können, benötigt der Spieler mehr woogoo als er durch Aktionen im Spiel sammeln kann. Er wird zwar immer in der Lage sein, einige Produkte an Roberta zu verkaufen. Möchte er aber mehr Geschäfte machen, muss er entweder seine Freunde um Hilfe bitten oder bezahlen. Die Verbrauchsgüter helfen auf diese Weise einerseits dabei, dass die Spieler schneller vorankommen, andererseits verlängern sie jede einzelne Spielsession.

Der Report Guy

Wie bereits erwähnt, bedarf es vieler Zahlen, die täglich analysiert werden müssen, um ein Spiel wie Monster World stetig zu verbessern. Wir befinden uns in der privilegierten Situation, die Zahlen von mehr als 1,8 Millionen Spielern täglich anschauen zu können – und das tun wir auch. Eine von wooga gebaute Reporting Website gibt uns einen wichtigen Einblick in unsere Spiele. Unser »Report Guy« schickt uns täglich automatisierte Report-Mails über zuvor definierte Zahlen. Abseits von all den Zahlen jedoch existiert eine ganze Reihe von Faktoren, die

Abbildung 8: Die Spieler können das begehrte **woogoo** in unterschiedlichen Paketen kaufen.

nicht auf Zahlen basieren, aber dennoch einen wichtigen Teil zur Erfolgsgeschichte von Monster World beigetragen haben.

Usability Testing

Seit Beginn der Entwicklung laden wir Spieler von außerhalb des Projektteams ein, um an Usability-Tests bei wooga teilzunehmen. Mein erster Test für Monster World basierte auf einem Papierprototyp und einem Stift, der als Maus fungierte. Ich zeigte den Prototyp möglichen Spielern und bat sie darum, den Stift zum »Klicken« zu benutzen. Die Seite nach dem Klick war als weiterer Papierprototyp bereits vorbereitet. Durch das häufige und vor allem frühe Testen konnten wir konzeptionelle Fehler vermeiden, bevor wir mit der eigentlichen Entwicklung begannen.

Später haben wir an richtigen Computern getestet. Die meiste Zeit haben wir unsere Test-Spieler einfach nur beobachtet. Dadurch, dass wir ihnen keine Fragen beantwortet und sie nicht durchs Spiel geleitet haben, wurden uns viele Designfehler der aktuellen Version bewusst. Das viel beschworene »We listen to our users!« wurde zum »We look at our users!«

Wir testen unsere Spiele alle zwei Wochen. Die Produktverantwortlichen sind jedes Mal anwesend. Direkt nach dem Test diskutieren die »Beobachter« ihre Funde und entscheiden über die Prioritäten für die nächste Iteration. Regelmäßig laden wir zusätzlich Teammitglieder ein, an den Tests teilzunehmen. Besonders für Ent-

Abbildung 9: Nicht nur das Spiel selbst wird **lokalisiert**, auch virtuelle Güter und die Spielwelt inklusive der Charaktere erhalten in manchen Fällen (wie der Fußball-WM) ein neues Aussehen.

wickler und Grafiker sind diese Sessions sehr wichtig, um einen Einblick in die Welt des »echten« Spielerverhaltens zu bekommen.

Release Cycles

Wir arbeiten in wöchentlichen Entwicklungszyklen und mit einer Mischung, bzw. dem Besten aus Scrum und Kanban. Jeden Dienstag veröffentlichen wir eine neue Version unseres Spiels auf Facebook. Der nächste Zyklus beginnt am Dienstag und endet am Freitag. Über das Wochenende können wir dann zur Ruhe kommen und uns darauf vorbereiten, am Montag die gefundenen Bugs zu fixen.

Diese kurzen Iterationen sind hart und jedes Teammitglied muss sehr diszipliniert arbeiten, um Verzögerungen bedingt durch Abhängigkeiten zwischen den einzelnen Teammitgliedern zu vermeiden. Aber es zahlt sich aus. Wir bewegen uns schnell und sind flexibel genug, um Prioritäten im Laufe des Tagesgeschäfts neu zu setzen. Sobald ein Feature fertiggestellt ist, veröffentlichen wir es in der Regel am gleichen Tag. Trotzdem behalten wir den wöchentlichen Rhythmus bei. Wenn also jemand im Team eine Idee hat (oder eine Erkenntnis aus einem Usability Test), kann dies bereits in die nächste Version einfließen – falls die Priorität hoch genug ist. Aus meiner Erfahrung kann ich sagen, dass dieser direkte Einfluss auf das Spiel das Team sehr stark motiviert.

Abbildung 10: Die **emotionale Bindung** zu den Charakteren erhöht sich besonders, wenn man die Früchte der eigenen Arbeit bewundern kann. Wie zum Beispiel dann, wenn man dem Monster-Baby ein Schaukelpferd gebaut hat.

Lokalisierung

Wir lokalisieren unsere Spiele in sieben Sprachen: Englisch, Spanisch, Italienisch, Französisch, Portugiesisch, Türkisch und Deutsch. Von den ersten Tagen der Entwicklung an haben wir unterschiedliche Sprachen im Spiel via XML unterstützt. Akzente zu unterstützen ist genauso wichtig wie eine Problemlösung für die Tatsache, dass manche Sprachen einfach mehr Platz benötigen. Die Folge ist, dass kein Text direkt in Bilder eingebettet ist. Wir haben bei wooga außerdem einen muttersprachlichen Kundensupport, lokalisierte Fan-Seiten und länderspezifische virtuelle Güter. Zum Beispiel haben wir während der Fußballweltmeisterschaft französische, italienische und spanische Trikots von Wäscheleinen im Spiel hängen lassen.

Emotionen

Ich hatte bereits erwähnt, dass das Spieler Engagement steigt, je mehr die Spieler eine Beziehung zu den Charakteren aufbauen. Wir konzentrieren uns sehr auf viele kleine Details, die Monster World liebenswürdig und ein bisschen verrückt machen. Während Robert der Roboter im Garten auf Aktionen des Spielers wartet, springt er zum Beispiel Seil, spielt PSP oder flirtet mit Roberta. Die Spieler können außerdem die Gärten ihrer Freunde besuchen und Robert dort auf eine Flasche Öl einladen. Wenn Robert zu viel Öl intus hat, wird er betrunken und verrechnet sich beim nächsten Deal. Dann zahlt er ein paar Münzen extra.

Es gibt eine ganze Reihe unterschiedlicher Charaktere in der Monsterfamilie, das vorhin erwähnte Baby eingeschlossen. Die Frauen in unseren Usability-Tests haben oft ein Leuchten in den Augen, wenn es auftaucht und bauen dann umso motivierter ein Schaukelpferd.

Das Team

wooga ist in Spiele-Teams mit jeweils einem Product Lead organisiert. Alle Teammitglieder – Entwickler, Grafiker, Projektmanager und zusätzliche Produktmanager – sind einem einzelnen Spiel zugehörig. Jedes Team besitzt außerdem einen eigenen Bürobereich, der für kurze Kommunikationswege sorgt.

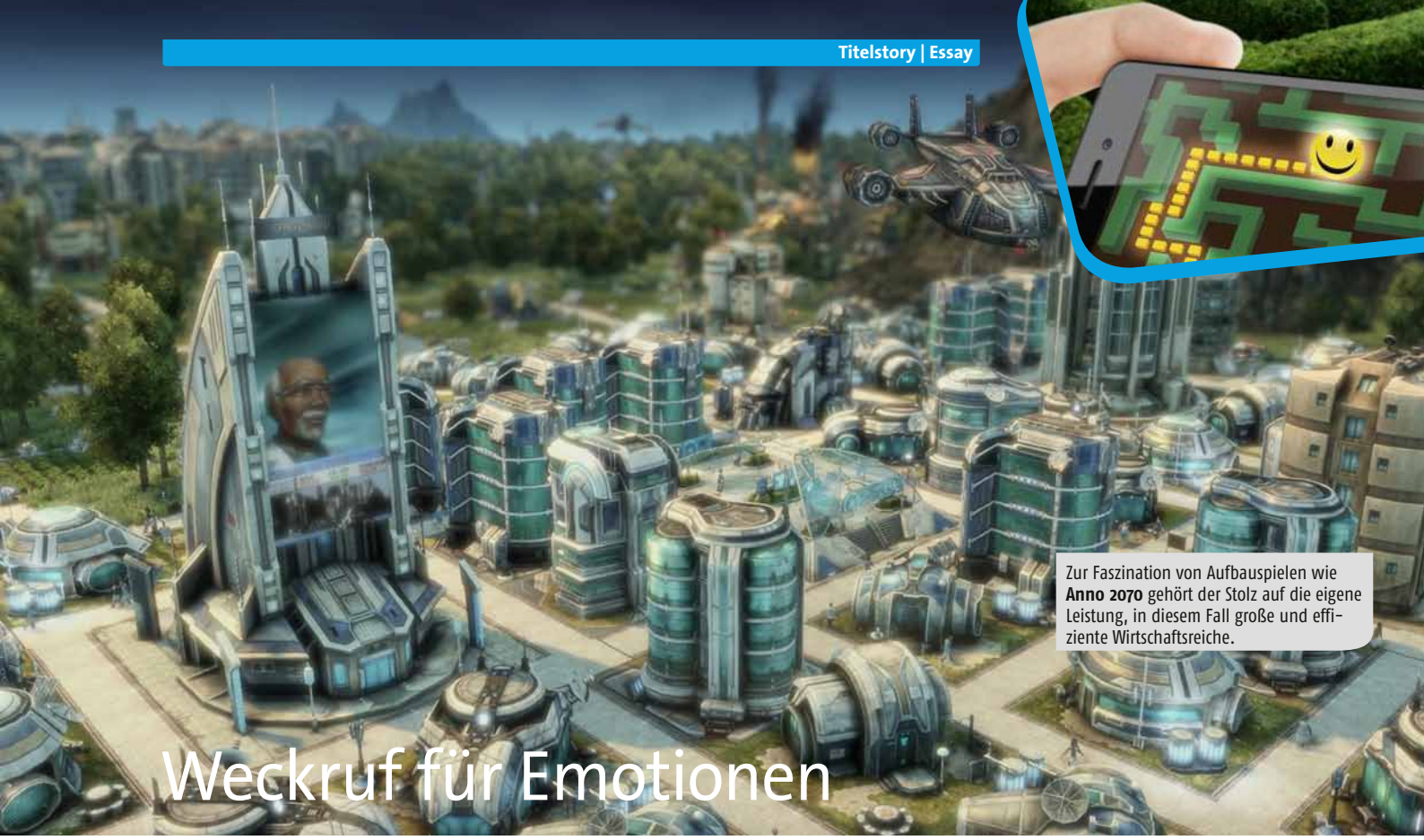
wooga ist in der glücklichen Lage, Menschen gefunden zu haben, die mit ganzem Herzen an einem gemeinsamen Ziel arbeiten: das beste Produkt, das sie sich vorstellen können. So arbeitet jedes Monster in unserem Team mit Enthusiasmus daran, das Spiel in eine echte Monster World zu verwandeln.

Stephanie Kaiser

Quellenverweis

Der Artikel »How to Care about Your Monsters« erschien im Sommer 2011 im Casual Connect Magazine und wurde für Making Games übersetzt und aktualisiert.

casualgamesassociation.org/magazine.php



Zur Faszination von Aufbauspielen wie **Anno 2070** gehört der Stolz auf die eigene Leistung, in diesem Fall große und effiziente Wirtschaftsreiche.

Weckruf für Emotionen

Spielspaß erfühlen

Was in Spielern vorgeht, wird heute erforscht, beobachtet und berechnet. Lässt sich Spielspaß auf diese Weise ausloten? Wer ein vergnügliches Spiel erschaffen will, sollte sich ebenso fragen, welche Gefühle er mit diesem auslöst.

Um das Jahr 161 nach Christus ging der griechische Arzt Galenus von Pergamon in Rom dem Geheimnis der Lebenskraft nach: Irgendwo musste die doch zu finden sein im Körper! Weil in Rom Experimente an Menschen verboten waren, untersuchte Galenus im Kolosseum verwundete Gladiatoren, parallel sezierte er Schweine und Hunde bei lebendigem Leib. Schließlich stand fest: Die Lebenskraft, pneuma psychikon, musste als Substanz im Inneren der Nervenbahnen durch den Körper pulsieren. 1.400 Jahre lang blieb das der Weisheit letzter Schluss.

Die Idee der Lebenskraft ist längst entkräftet. Heutzutage hat eine heftige Suche nach einem anderen Wunderding begonnen: dem Spielspaß. Schweine und Hunde bleiben glücklicherweise verschont, aber am Menschen wird umso emsiger geforscht. Neuropsychologen schauen ihm in den Kopf, Psycho- und Soziologen studieren seine kognitiven Prozesse und sprechen über Flow-Erleben, Erfahrung von Selbstwirksamkeit, Feedback-Schleifen und Immersion im virtuellen Raum. Analysten und Statistiker spicken ihm auf die Hände, sie vermessen sein Verhalten: Wie lange spielt er, auf welcher Stufe gibt er auf, wie oft pro Stunde öffnet er den Ingame-Shop, und wie lange starrt er die Preise an, bis er das Fenster wieder schließt, ohne etwas gekauft zu haben? In der Welt des Data Minings ist Spielvergnügen quantifizierbar über Zeiten, Klicks und Datenströme.

Und dann ist da noch der Spielekritiker, der dem flüchtigen Spielspaß qua Beruf nachstellt, um ihn dann so exakt festzunageln, dass Sozialwissenschaftler nur neidisch mit dem Klemmbrett wackeln können: 85 von 100 Punkten! Da die meisten Spieletester mit Schweinen nur in Schnitzelform interagieren, noch nie irgendjemanden in die MRT-Röhre geschoben haben und den Umgang mit echten Spielern nach Möglichkeit vermeiden, darf man sich durchaus fragen: Wie machen die das? Wie erkennen sie Spielspaß, wie definieren sie ihn?

Herz des Testers

Zwar ziehen die meisten Spieletester Checklisten oder zumindest grobe Kategorien zurate, um ein Spiel in grundlegende Elemente aufzuteilen und sie dadurch vergleichbar, mithin objektivierbar zu machen – die Qualität der Grafik, die Funktionalität der Steuerung etc. Wobei der Idealfall für die Steuerung interessanterweise zu sein scheint, dass sie den Spielspaß nicht beeinträchtigt, ein intuitives Interface Spaß also

Christian Schmidt

ist Market Research Analyst bei Bigpoint.

Christian arbeitete 13 Jahre lang als Spielejournalist für GameStar, zuletzt als stellvertretender Chefredakteur. Seit Dezember 2011 ist er Market Research Analyst für Bigpoint in Hamburg. Spaß macht ihm eigentlich kein Spiel, sagt er. Außer die von Bigpoint, die seien wirklich exzellent. Christian ist zu erreichen unter: cs216@web.de

nur ermöglicht, aber nichts zu ihm beiträgt – ein Gedanke, der in Zeiten von Wii, Move und Kinect, aber auch der Wisch-und-tipp-Steuerung von Smartphones und Tablets veraltet ist, wo allein die Bedienung ohne jeglichen Spielinhalt bereits körperliches Vergnügen auslösen kann. Aber in der Regel ist der wichtigste Untersuchungsgegenstand jedes Spieletesters immer noch er selbst, und sein zentrales Instrument die gute alte Introspektion. Im Mittelpunkt seiner Betrachtung steht nicht die kognitive und nicht die quantitative, sondern die emotionale Komponente des Spielspaßes – nicht Kopf und Hand also, sondern das Herz, und Spielspaß als komplexes Konglomerat von Gefühlszuständen.

Warum macht es eigentlich Freude, bei Tetris abstrakte Blöcke zu Reihen zu sortieren? Und ist diese Art von Spaß die gleiche, die jemand empfindet, der bei Team Fortress 2 fünf Gegner in Folge ausgeschaltet hat? »Das hat Spaß gemacht!« ist eine Bilanz, eine positive Summe aus Einzelempfindungen, die aus unterschiedlichen Quellen entspringen und in Teilen auch negativ sein können, sogar sein sollten. Wer als Spieldesigner Spaß entstehen lassen will, verfolgt ein Fernziel. Der Weg dorthin führt über verschiedene Emotionen, die zu erzeugen potenziell einfacher ist als die High-Level-Idee des Spielspaßes.

Stolz ermöglichen!

Als eines der grundlegenden Gefühle in Spielen entsteht Stolz – also Zufriedenheit mit der eigenen Leistung – aus einer großen Menge unterschiedlicher Situationen. Dabei ist gleichgültig, welche Art von Leistung zum Ergebnis geführt hat – das kann ein außergewöhnlicher Skillshot in Battlefield 3 (Geschicklichkeit) ebenso sein wie der im zehnten Anlauf endlich geschaffte Boss in God of War 3 (Lernprozess) oder die über 100 Stunden sorgsam aufgebaute Stadt in Anno 2070 (Ausdauer). Wichtig ist nur, dass es eine Leistung war, also ein Widerstand zu überwinden, eine Anstrengung notwendig war, und dass dem Ergebnis ein Wert innewohnt. Stolz bedeutet, sich selbst als kompetent zu erkennen. Jedes Spiel tut gut daran, diese Erkenntnis durch Belohnungen zu unterstützen. Klassische Ranglisten und moderne Achievements sind mächtige Hilfsmittel, aber entscheidend bleibt letztendlich, dass es Hindernisse geben muss, die man überwinden kann, indem man sich Kompetenz erwirbt; kognitiv (durch Regelbeherrschung), motorisch (durch geschickte Steuerung) oder schlicht durch Sitzfleisch.

Neugier wecken!

Spieledesigner dürfen Pandora danken, dass aus ihrer Büchse auch die Hoffnung schwirrte. Selbst die langweiligste Routineklickerei wird erträglich, wenn in ihr Rätsel angelegt sind, auf deren Aufklärung man hofft. Neugier darauf, was als Nächstes kommt, ist eine extrem mächtige Motivationsquelle. Sie entsteht sowohl aus der Spielmechanik, vor allem aus Levelsystemen (»Welches Talent bekomme ich auf der nächsten Stufe?«, »Welches Gebäude wird als Nächstes freigeschaltet?«, »Wie wird das nächste Gebiet aussehen?«), als auch aus der Narration, also den Geheimnissen der Story (»Wer ist der Mörder?«, »Werde ich Miranda ins Bett kriegen?«). Die Diablo-Serie ist auch deshalb ein so mustergültig motivierendes Spiel, weil sie in schneller Taktung spielmechanische Neugier erzeugt, auf den nächsten Gegenstand, den nächsten Dungeon, das nächste freigespielte Talent.

Neugier zu erzeugen bedeutet, dem Spieler bewusst zu machen, dass er mit etwas Neuem rechnen kann, wenn er weiterspielt. Das können selbst implizite Elemente sein wie die amüsanten Slapstick-Animationen der Sims-Serie, von

Tali: Ich habe gerade eine Nachricht von der Migranten-Flotte erhalten. Der Admiralitätsrat beschuldigt mich des Verrats.

Gute Geschichten wie in Mass Effect 2 motivieren durch starke Emotionen: Sympathie, Mitgefühl, Zorn, vor allem aber **Neugier** darauf, wie es weitergeht.

denen man als Spieler so viele wie möglich sehen möchte und sich über jede frisch entdeckte freut. Allerdings ist Neugier insofern ein zweischneidiges Schwert, als sie Spieler dazu bringt, sich auch durch langweilige Passagen zu quälen. Am Schluss wird gegengerechnet: War das Ergebnis den Aufwand wert? So entscheidet sich rückwirkend, ob die vergangene Stunde spaßig oder letztendlich frustrierend war, weil die dumme Kuh nach all dem Gebagger nicht mal nackte Haut zeigt.

Ungewissheit ermöglichen!

Das Wissen, dass etwas passieren wird, aber die Unsicherheit darüber, was genau – das ist Spannung. Weil der Spieler im interaktiven Computerspiel selbst handelt, trifft er Entscheidungen und muss letztlich auch die Konsequenzen tragen. In den Worten Sid Meiers bedeutet gutes Spieldesign also, den Spieler regelmäßig »vor interessante Entscheidungen zu stellen«. Jeder Zug im Schachspiel ist eine solch interessante Entscheidung, weil in ihm immer ein Grad von Ungewissheit angelegt ist: War das eine gute Wahl? Wie wird mein Gegner reagieren? Spannende Spiele lassen den Spieler zumindest eine Zeit lang im Unklaren darüber, ob seine Entscheidung gut oder schlecht war, aber sie legen die Auswirkungen irgendwann nachvollziehbar dar, um ihm eine Selbstbewertung zu ermöglichen. Spannende Unsicherheit besteht indes auch darin, nicht zu wissen, was passiert wäre, wenn man sich anders entschieden hätte. Im Unterschied zum realen Leben kann man das in Spielen herausfinden, indem man die Situation wiederholt – der klassische Wiederspielwert.

Pflichtgefühl, Rachsucht und Schadenfreude forcieren!

In wettbewerbsorientierten Spielen wie Multiplayer-Shootern konzentriert sich die Spielmechanik in den entscheidenden Momenten auf die Frage, wer von zwei Kontrahenten im Zweikampf die Oberhand behält. Für den Sieger ist eine kurze Stolzspitze ein starker Teil des Vergnügens. Aber dieses Kräftemessen macht im besten Fall selbst dann Spaß, wenn man unterliegt, weil sich das Duell in kurzer Folge wiederholen lässt. Die soziale Dynamik beim Spiel zwischen wildfremden Menschen lässt spontane emotionale Bindungen entstehen, die starke Wirkung entfalten können: Dankbarkeit gegenüber dem Medic, der mich gerade in letzter Sekunde gerettet hat, Pflichtgefühl gegenüber den Squad-Kameraden, Rachsucht gegenüber dem Sniper, der einen jetzt schon zum zweiten Mal vor dem Torbogen erschossen hat. Und vor allem Schadenfreude als Erleichterungsreaktion, die umso befriedigender ausfällt, je höher vorher die Spannung war, etwa wenn man jenen vermaledeiten Sniper zehn Sekunden vor Partieende doch noch messert. Solche Gefühle kommen ins Spiel, sobald ein direkter Wettbewerb zwischen Spielern stattfindet. Und sei es nur, die Rollenspiel-Party am Ende des Dungeons um den Inhalt der Schatztruhe würfeln zu lassen.

Ordnungsliebe unterstützen!

Eine der interessanteren Eigenschaften von Menschen ist, dass ihnen das Schaffen von Ordnung ein Gefühl von Befriedigung gibt. Die Suche nach Mustern und die Abwehr von Unsicherheit sind in unserem Denken angelegt, entsprechend belohnt uns das Gehirn für die Schaffung von Strukturen. Eine Vielzahl der populärsten Gesellschaftsspiele schlägt daraus Kapital, von Memory bis Puzzles, von Patience bis Mahjongg. Auch Videospiel-Klassiker wie Tetris oder das Kistenschiebespiel Sokoban machen das Ordnen von unordentlichen Situationen zum hauptsächlichen Spielinhalt. Aber selbst komplexere Genres enthalten fast immer ein Ordnungselement – jeder Rollenspieler kennt die Zufriedenheit, wenn er ein vollgemülltes Inventar ausgemistet hat, jeder Aufbauspieler plant seine Städte nach einem inneren Modell. Zu dieser Ordnungsliebe gehört auch der Wunsch nach Abschluss, nach dem Beenden einer offenen Aufgabe. Wie stark dieser Drang sein kann, beweisen etwa Rollenspiele mit zahlreichen Nebenquests, bei denen allein der »Ach, das mach ich noch schnell«-Effekt Partien bis in die frühen Morgenstunden ausdehnen kann. Aber auch Social Games der Zynga-Bauart setzen stark auf das Konzept, den Spieler zu jedem Zeitpunkt immer gut sichtbar (!) mit mehreren offenen Aufgaben zu konfrontieren, und seien sie noch so albern. Der Wunsch, offene Ziele abzuschließen und die Auftragsliste damit etwas ordentlicher zu gestalten kann so stark sein, dass die Spieler in Free2Play-Modellen dafür sogar Geld ausgeben, einfach um das Ding endlich vom Tisch zu haben.

Gefühle verstehen!

Grundlegende Emotionen wie diese sind verwandt und verschränkt mit zahlreichen weiteren – mit Ehrgeiz und Geltungsdrang, Mitgefühl und Geborgenheitswunsch, auch mit Wut und Ärger. Entscheidend ist der Gedanke, dass Spaß unterm Strich die Summe von Eindrücken ist, die den Spieler bewegt haben, die ihn aus der Gleichgültigkeit reißen. Wer ein gutes Spiel machen will, der kann in Regeln und Abläufen denken und in Verhaltensdaten und Statistiken. Er kann sich aber auch fragen, was der Spieler zu welchem Zeitpunkt fühlt. Das geschickt zu lenken ist eine der Herausforderungen von wirklich gutem Spieldesign.

Christian Schmidt

Im rechten Moment der rächende Schuss und Schadenfreude über das Ableben des Gegners – eine Kernmotivation in Multiplayer-Shootern wie **Battlefield 3.**

Warum ist ein simples Spiel wie **Windows Solitaire** so motivierend? Ein Grund ist das befriedigende Spielprinzip: Ordnung in Unordnung zu bringen.

Social Games wie **Empires & Allies** bedienen den Abschlussdrang der Spieler durch Miniquests (Spalte am linken Bildschirmrand).

Mit Formeln zum perfekten Balancing
Spielspaß kalkulieren

Nicht nur durch pures Ausprobieren lässt sich Spielspaß herbeiführen, sondern er lässt sich auch theoretisch begründen. Wie genau das funktioniert, erklärt Game Designer Daniel Dumont anhand konkreter Formelbeispiele.

Daniel Dumont

ist Mitgründer und Creative Director der Gaming Minds Studios.

Daniel ist seit 13 Jahren Spieleentwickler. Er entwickelte bei Ascaron die Patrizier-Reihe weiter und entwarf Port Royale und Darkstar One. Im Jahr 2009 gründete er zusammen mit Kay Struve und Kalypso Media die Gaming Minds Studios, die kürzlich einen neuen Teil der Patrizier-Reihe veröffentlicht haben und nun an einer Fortsetzung von Port Royale arbeiten. Daniels Arbeitsschwerpunkt liegt in der Visionierung sowie in der Konzeption von Computerspielen.

Hört sich doch erst einmal komisch an, Spielspaß kalkulieren zu wollen. Ein Spiel besteht aber nun mal aus vielen Zahlen, Parametern und Formeln. Und wenn eine Spielmechanik Spaß macht, scheinen diese ja zu stimmen – sie sind untereinander ausbalanciert. Nun gibt es prinzipiell zwei Möglichkeiten, ein Spiel zu balancen: eher praktisch oder eher theoretisch. Praktisch heißt, das Spiel wird immer wieder gespielt und getestet; bei jedem Durchgang werden diverse Werte angepasst, um den Spielspaß weiter zu erhöhen. Theoretisch heißt, man überlegt sich bereits in der Konzeptionsphase anhand ausgedachter Spielsituationen, wie die entsprechenden Werte zueinander ausgeglichen werden müssen.
Weder nur das eine noch allein das andere gibt es im echten Leben. Um praktisches Balancing kommt man nicht herum. Man kann das theoretische Balancing jedoch sehr weit treiben und dadurch die später anstehenden Balancing-Tests erheblich verkürzen.

In diesem Artikel möchte ich mit relativ einfachen und konkreten Beispielen zeigen, wie sich Berechnungen im Vorfeld durchaus positiv auf den Spielspaß auswirken können. Das letzte Beispiel beschreibt einen Fall, der sich durch Ausprobieren gar nicht mehr balancen lässt, sondern eben nur noch theoretisch. Womit wir dann vollends beim kalkulierten Spielspaß wären.

Um den Spielspaß eines Seekampfes wie hier in **Port Royale 3** zu kalkulieren, sollte man sich zunächst die Verbindung der zwei zentralen Parameter Kampfdauer und Ladezeit der Kanonen vor Augen führen.

Beispiel 1: Finden von Formeln und Werten

Für den Anfang habe ich mir ein kleines Spiel ausgedacht, dessen Formeln und Parameter sich mit den richtigen Überlegungen bestimmen lassen, ohne das Spiel selbst zu spielen. Es handelt sich um eine Variante des alten C64-Spiels Space Taxi (siehe **Abbildung 1**).

Es gibt insgesamt fünf Plattformen. Auf jeder kann ein Fahrgast erscheinen, der zu einer anderen Plattform möchte. Mit dem Taxi holt man ihn ab und bringt ihn zu seinem Ziel. Die Schwierigkeit dabei: Um das Taxi zu bewegen, setzt man Schubdüsen ein, die einen begrenzten Treibstoffvorrat reduzieren. Natürlich darf man nicht zu schnell auf einer Plattform aufsetzen. An der mittleren Plattform wird das Taxi aufgetankt. Je schneller man den Fahrgast abliefert, desto mehr Fahrtgeld erhält man. Folgende Parameter sind die wichtigsten, die es in diesem Beispiel zu ermitteln gilt:

- Maximalgeschwindigkeit
- Beschleunigung und Gravitation
- Spritvorrat, Spritverbrauch

Beschleunigung und Gravitation

Für die vertikale Raumrichtung gilt für die Geschwindigkeitsänderung »dv« innerhalb eines Zeitintervalls »dt«:

$$dv = a \times dt - g \times dt$$

Und für den zurückgelegten Weg »ds« gilt:

$$ds = 0.5 \times (a - g) \times Zeit^2$$

Dabei ist »a« die Beschleunigung durch die Schubdüsen nach oben und »g« die Gravitati-

onsbeschleunigung nach unten. Wie kommt man nun auf die Werte »a« und »g«?

Hierzu überlegt man sich am besten eine Extremsituation, die genau so im Spiel vorkommen kann und daher realistisch ist. Ich stelle mir also vor, mein Taxi befindet sich am unteren Bildschirmrand und ich betätige die Hauptdüsen, bis das Taxi den oberen Rand erreicht. Nach welcher Zeit soll das sein? Weil das Taxi erst langsam steigt und dann immer schneller wird, gehe ich mal von 2 Sekunden aus. Wenn ich das Taxi vom oberen Rand herunterfallen lasse (also ohne Schub), sind 2 Sekunden ebenfalls okay, denke ich.

Wenn man nun annimmt, das Bild habe eine Größe von 800x600 Pixeln (die Größe ist egal, sie lässt sich später einfach anpassen), so kann man mithilfe der beiden Situationen und der zweiten Formel folgende Werte leicht ermitteln:

a = 600 und g = 300

Als Nächstes fällt auf, dass die Maximalgeschwindigkeit nicht relevant ist, denn welche Geschwindigkeit das Schiff nach 2 Sekunden erreicht hat, ist weniger wichtig als die Tatsache, dass es nach dieser Zeit vollen Schubes aus dem Bild verschwindet.

Spritvorrat und Spritverbrauch

Auch diese Parameter lassen sich leicht bestimmen. Wenn man davon ausgeht, dass der Schub zur Seite etwa so groß ist wie die Gravitation (betätige ich eine seitliche Schubdüse, so beschleunigt das Taxi genauso schnell, als wenn es von oben herunterfallen würde), dann könnte man sich folgende Frage stellen: Wie lange kann ich zwischen linkem und rechtem Rand hin- und herfliegen, bevor der Sprit zur Neige geht und ich auftanken muss?

Während dieser Zeit muss ich die Schubdüsen nach oben zur Hälfte betätigen, von den Seitenschubdüsen ist außerdem immer eine aktiviert (bewegen und bremsen). Der Schub nach oben benötigt pro Sekunde 0.5 x Verbrauch x 2 (weil die Schubdüsen doppelt so kräftig sind), der Schub zur Seite benötigt 1 x Verbrauch.

Wie lange soll ich nun fliegen können? 5 Sekunden sind zu wenig, denn dann muss ich ja permanent tanken und schaffe es kaum, einen Fahrgast zu transportieren. 30 Sekunden sind wiederum eher zu viel, weil ich dann den Tank kaum im Auge behalten muss. Also scheint mir ein Wert von 15 oder 20 Sekunden sinnvoll. Wenn ich den Tankinhalt nun willkürlich auf 1.000 Liter setze, gilt:

1.000 Liter = (0.5 × Verbrauch × 2 + 1 × Verbrauch) × Zeit

Daraus folgt wiederum: Verbrauch = 25 Liter pro Sekunde für eine Seitendüse und doppelt so viel für die große Schubdüse.

Zusammenfassung

Dieses kleine Beispiel sollte zeigen, wie man Berechnungen durchführen kann, indem man sich eine im Spiel vorkommende Situation vorstellt und die relevanten Parameter so berechnet, dass der Spieler den meisten Spaß dabei hat.

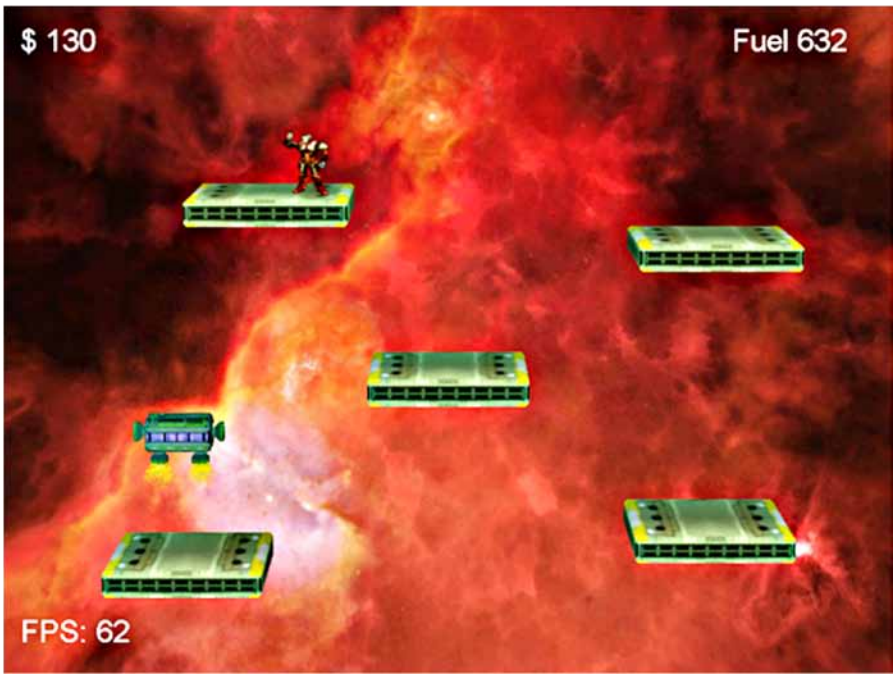


Abbildung 1: Bei diesem Bild handelt es sich um eine nachgestellte Szene aus dem C64-Klassiker Space Taxi. Dieses einfache Beispiel lässt sich **vollständig theoretisch balancen.**

Ein Tipp: Spezielle oder extreme Situationen kann man sich häufig vor dem geistigen Auge besser vorstellen; der Zusammenhang der Formeln und Parameter ist leichter zu durchschauen. Der Vorteil ist, dass die Werte, die man für solche Situationen findet, auch für die allgemeineren Fälle gelten und daher eine solide Basis für das spätere praktische Balancing bieten.

Beispiel 2: Wertentwicklungen

Wenn man es im Spiel mit Entwicklungen von Werten zu tun hat – zum Beispiel mit dem Aufstieg des Spielers und damit zusammenhängenden Werten – so lässt sich Spielspaß kalkulieren, indem man der Wertentwicklung eine mathematische Kurve mit passendem Verlauf zugrunde legt. Hierzu möchte ich im Folgenden ein Beispiel geben, zwei weitere finden Sie als Kurven auf den nächsten Seiten.

Die Entwicklung der Stärke

Wir stellen uns vor, der Spieler steigt in einem Action-Rollenspiel von Level 0 bis Level 20 auf und wird entsprechend stärker. Gleichzeitig dringt er in immer schwierigere Regionen der Spielwelt vor und die Gegner werden mächtiger und zahlreicher. Wenn man nun wüsste, wie sich die Stärke des Spielers entwickelt, könnte man besser planen, wie sich die Gegner entwickeln sollen. Zur Vereinfachung betrachten wir im Folgenden nur die Stärke der Waffen des Spielers. Die Frage ist, in welchem Maße die Stärke der Waffen zunehmen soll, sodass der Spieler etwas davon merkt, es für ihn aber auch nicht zu einfach wird. Eine Möglichkeit ist, die ideale Waffenstärke anhand einer Kurve festzulegen. Egal wie viele Waffen es später geben

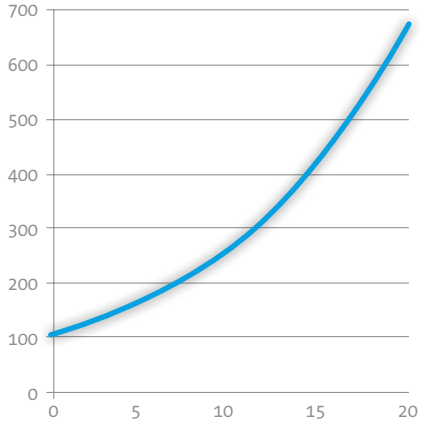


Abbildung 2
Exponentialkurve: Stärke = 100 × (1 + 10%) ^ Level
Sie eignet sich für die Zunahme der Stärke des Spielers. Die Kurve stellt sicher, dass der Spieler stets einen Stärkezuwachs wahrnimmt, wenn man die Stärke einer Waffe, eines Schildes, einer Rüstung etc. an den Kurvenpunkten orientiert.

wird, solange sie sich an dieser idealen Stärke orientieren, folgen sie im Mittel der Kurve.

Nehmen wir an, wir beginnen mit einem willkürlichen Wert der Stärke 100. Wird pro Level die Waffenstärke im Schnitt um 10 zunehmen, ist das anfangs in Ordnung, denn der Spieler kann es spüren, weil bekannte Gegner deutlich schneller beseitigt sind. Hat die Waffe jedoch eine Stärke von 200 erreicht, ist 10 nicht mehr so viel. Besser ist es also, wenn die Zunahme ebenfalls steigt. Dies erreicht man leicht mit einer Exponentialfunktion, wie sie jeder von Bankzinsen her kennt (**Abbildung 2**). Wenn wir festlegen, dass die Waffenstärke nicht um 10, sondern um 10 Prozent steigt, so bleibt sie für den Spieler immer spürbar. Die Formel lautet:

```
Waffenstärke = 100 × (1 + 10%) ^ Level
```

Bei der Festlegung der Waffenstärke orientiert man sich nun an der Kurve. Wenn der Spieler beispielsweise eine Level-10-Waffe besitzt, so hat sie die Stärke 260. Findet er eine Level-12-Waffe, so hat sie bereits die Stärke 314 usw.
Der Effekt lässt sich noch weiter steigern, indem man Werte zurückhält. Verteilt man zum Beispiel nur eher schwache Level-5-Waffen im Spiel und macht die Level-6-Waffen stärker, spürt der Spieler einen deutlichen Sprung von Level 5 auf 6, was sich letztlich als Abwechslung im Spiel niederschlägt. Einen ähnlichen Effekt erzielt man, wenn der Spieler nur alle zwei Levels eine neue Waffe erhält.

Indem man sich für eine Wertentwicklung, wie sie in einem bestimmten Feature vorkommt, einen geeigneten Verlauf überlegt, gelangt man sehr schnell an passende Werte für das Balancing. Natürlich sind noch viele andere Kurven denkbar, etwa quadratische oder zusammengesetzte Kurven mit unterschiedlichen Verläufen in verschiedenen Abschnitten.

Beispiel 3: Das Ausbalancieren von zusammenhängenden Werten

Nehmen wir an, wir müssten eine Seeschlacht mit Segelschiffen im Vorfeld kalkulieren. Dann würde man sich zunächst überlegen, welche Werte hier in irgendeiner Form eine Rolle spielen. Das wären:

- Schaden pro Kanone (SK)
- Anzahl Kanonen (AK)
- Trefferpunkte eines Schiffs (TP)
- Ladezeit der Kanonen (LK)
- Geschwindigkeit der Schiffe
- Beschleunigung der Schiffe
- Drehgeschwindigkeit der Schiffe

Nun sind die letzten drei Parameter von anderer Natur als die ersten vier. Denn sie beschreiben die Bewegung der Schiffe und haben eher etwas mit Spielgefühl zu tun. Mit diesen Parametern würde ich daher beim Balancen so vorgehen wie in Beispiel 1. Die ersten vier Parameter jedoch lassen sich prima zusammen betrachten.

Man stellt sich zunächst wieder eine einfache Extremsituation vor. Zwei ähnlich starke Schiffe stehen sich regungslos gegenüber und feuern eine Breitseite nach der anderen auf ihren jeweiligen Gegner ab. Bei solchen Situationen wird schnell klar, dass alle Parameter zusammen gebalanct werden müssen. Denn es ergibt schließlich keinen Sinn, den Schaden einer Waffe zu bestimmen, wenn man nicht auch die Trefferpunkte dazu in Bezug setzt. Aber wie kommt man nun auf den Spielspaß?

Die Verbindung zum Spielspaß für diese einfache Extremsituation ist die Kampfdauer und wohl auch die Ladezeit, wie man sich sehr leicht vor Augen führen kann.

- Wenn die Kampfdauer sehr kurz (z.B. 5 Sekunden) ist, wird es garantiert hektisch und in vielen Fällen auch unfair. Ist sie zu lang (z.B. 5 Minuten), wird es langweilig.
- Ebenso langweilig wird es, wenn die Ladezeit zu lang ist, denn dann schaut der User nur auf den Kampf und lange Zeit passiert nichts. Wie kurz man die Ladezeiten wählt, hängt im Übrigen vom gewählten Setting ab. In einem historischen Setting wird man vermutlich um die 10 Sekunden wählen.

Wenn man nun die beiden Parameter festgelegt hat, kann man sich die fehlenden drei Parameter vornehmen. Wobei die Anzahl der Kanonen wohl eher wieder durch das Setting und die Optik der Schiffe vorgegeben wird. Der Zusammenhang der Größen ist gegeben durch:

```
TP = TP – AK × SK × Zeit / LK
```

TP' sind in diesem Fall die Trefferpunkte eines Schiffs nach einer gewissen Zeitspanne. Setzt man TP' = 0, so ist »Zeit« die Kampfdauer. Aufgelöst nach TP ergibt sich:

```
TP = AK × SK × Kampfdauer / LK
```

Wenn man hier die bereits bestimmten Parameter einsetzt, erhält man den Zusammenhang zwischen den Trefferpunkten und dem Schaden pro Kanone, der (zumindest theoretisch) den größten Spielspaß verspricht. Ob man diesen Schaden nun auf den Wert 1, 10 oder 100 setzt, spielt keine Rolle – das sind lediglich interne Werte. Ich empfehle, solche Werte aber möglichst klein (z.B. 1) zu wählen, damit sie später im Interface angezeigt werden können, ohne eine unnötige Zahlenflut zu erzeugen.

Nun könnte man denken, die Werte passten ja nur gut für diese Extremsituation. Das ist nicht ganz richtig, denn auch wenn die Schiffe später wild umherfahren, wird der User versuchen, am Ende der Ladezeit immer eine gute Schussposition zu haben. Und mit der Kalkulation haben wir damit eine realistische Grundlage geschaffen.

Beispiel 4: Balancing bei größeren Gesamtheiten

In diesem Beispiel würde ich stark davon abraten, die richtigen Werte durch Ausprobieren zu finden, denn es liegt ein Fall zugrunde, bei dem zu viele Objekte miteinander wechselwirken. Das Ausbalancieren eines Objekts würde unweigerlich das Balancing aller anderen Objekte der Gesamtheit wieder verschieben.

Um solche Fälle auszubalancieren, ist es erneut sehr wichtig, das Pferd von hinten aufzuzäumen. Man beginnt mit den Werten, die man eigentlich haben möchte und legt diese quasi fest. Anschließend rechnet man zurück, um auf alle verbleibenden Werte zu kommen, die dazu benötigt werden, das entsprechende Spielfeature vollständig zu beschreiben.

Diese Methode war sehr hilfreich bei der Konzeption der Wirtschaftssysteme in der Patrizier- und Port-Royale-Reihe. Und ich bin überzeugt, dass sie sich problemlos auf andere Wirtschaftssysteme übertragen lässt, in denen in irgendeiner Form eine Ressource erst produziert und dann verbraucht wird.

Hier ein konkretes, wenn auch vereinfachtes Beispiel: Wir stellen uns vor, in einer Spielwelt gibt es 10 unterschiedliche Waren, die ausschließlich von Arbeitern produziert werden. Ein Arbeiter produziert jedoch nur dann, wenn er mindestens 8 dieser Waren für den täglichen Verbrauch zur Verfügung hat, ansonsten wird er unzufrieden und haut ab. Außerdem gibt es einige Waren, die als Rohstoffe für andere Waren dienen. Wir haben es also mit Waren, Rohstoffen und Fertigwaren zu tun. Die Frage ist nun: Wenn man einen bestimmten Verbrauch der Arbeiter und eine bestimmte Produktion der Betriebe zugrunde legt, wie viele Betriebe benötigt man dann für ein stabiles System?

Stabil heißt, dass es Spielsituationen gibt, in denen Produktion und Verbrauch aller Waren einigermaßen im Gleichgewicht sind, denn weder eine permanente Überproduktion noch eine permanente Warenverknappung erzeugen Spielspaß. Interessanter ist vielmehr, wenn sich Verbrauch und Bedarf im Normalfall decken, durch diverse Ereignisse (wie z.B. den Ausbruch eines Feuers oder einer Seuche) allerdings gestört werden können.

Wer nach einer Lösung sucht, indem er mit Verbrauch und Produktion pro Betrieb herumprobiert und die Anzahl der Betriebe testet, stößt sehr schnell an Grenzen. Denn das gesamte System ist durch die Arbeiter mit ihrem konstanten Verbrauch miteinander verwoben. Wenn man das Ganze jedoch rückwärts kalkuliert, so wird es sehr viel einfacher. Hier würde ich wie folgt vorgehen:

1. Zunächst legt man fest, wie viel ein Arbeiter von jeder Ware verbraucht.
2. Nun legt man fest, wie viele Rohstoffe man für die benötigten Fertigwaren zusätzlich benötigt. Braucht ein Arbeiter beispielsweise pro Tag 2 Liter Bier und benötigt man dafür 0,5 kg Getreide, so steigt der Getreideverbrauch durch den Bierbedarf um 0,5 kg pro Tag.
3. Man trägt alle Waren in eine Tabelle ein und erhält dadurch für jede Ware pro Arbeiter einen bestimmten Tagesverbrauch.
4. Nun kommt der Clou: Spielspaß heißt Gleichgewicht (die Störungen kommen später dazu), d.h., das System muss so viel produzieren, wie es verbraucht; also verlangt man, dass die Tagesproduktion gleich dem Tagesverbrauch ist.

Dazu teilt man etwa 1.000 Arbeiter auf die 10 Betriebstypen auf und verlangt, dass die einem Betrieb zugeteilten Arbeiter genau die Waren produzieren, die alle 1.000 Arbeiter verbrauchen. Egal wie man die Arbeiter aufteilt, der Betrieb erzeugt dann genau die richtige Menge. Und das ist bereits die Lösung des Problems.

Und weil es so schön ist, weist man jedem Arbeiter einen Tageslohn zu und erhält sofort Preise für alle Waren und Rohstoffe. Mit Letzteren errechnet man dann die Preise für die Fertigwaren.

Fazit

Ich hoffe, die Beispiele haben gezeigt, dass es Spielsituationen gibt, die sich mithilfe einfacher Kalkulationen gut balancen lassen. Zudem kann man mit dem umgekehrten Balancing selbst große Parameter-Gesamtheiten in den Griff bekommen. Dabei lässt sich Beispiel 4 auch auf andere Spielsituationen übertragen, indem man die Begriffe »Produktion« und »Verbrauch« etwas weiter fasst. Wenn etwa in einem Echtzeit-Strategiespiel zunächst Einheiten produziert und anschließend »verbraucht« werden, um den Gegner zu schwächen, so lässt sich auch hier für bestimmte Bereiche ein umgekehrtes Balancing durchführen. Man könnte sich beispielsweise vorstellen, welche Spielsituation (etwa eine bestimmte Anzahl von Einheiten unterschiedlichen Typs) nach einer bestimmten Spielzeit vorliegen soll, damit der Spieler aller Voraussicht nach Spaß haben wird.

Doch dabei darf man nicht vergessen, dass es auf dem Weg zu dieser »kalkulierten« Situation Spielmerkmale wie Steuerung, Feedback und Überblick gibt, die mit der Kalkulation zwar nichts am Hut haben, für den Spielspaß letztlich aber ebenso wichtig sind.

Daniel Dumont

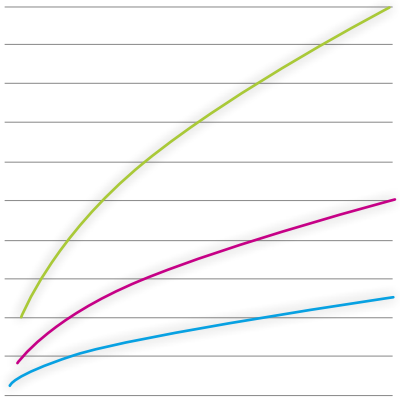


Abbildung 3
Wurzelfunktion: Stärke = A × Wurzel(X)
Nimmt anfangs stark zu, später weniger. Dieser Verlauf ist sinnvoll für die Zunahme von Möglichkeiten des Spielers, etwa wenn er Lernpunkte erhält oder eine Waffe upgraden kann, indem er die gleiche Waffe noch einmal findet und sie durch Zusammenschmieden verbessern kann. Die ersten Upgrades lohnen sich noch, später jedoch wird es immer sinnvoller, auf eine höherwertige Waffe umzusteigen, auch wenn diese anfangs schlechter ist.

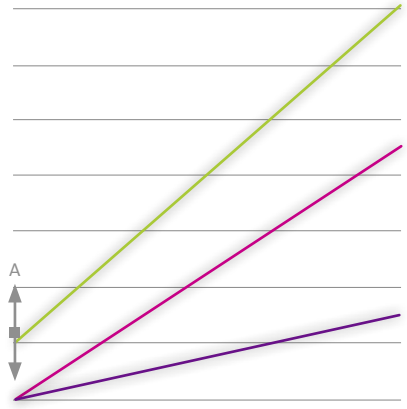


Abbildung 4
Linearer Anstieg: Y = A + B × X
Sie ist natürlich die einfachste Kurve und sozusagen der Mittelweg der beiden anderen. Man sollte sie immer dann verwenden, wenn nichts gegen einen linearen Anstieg spricht, wenn die Größe des Spielerinventars durch Upgrades zunehmen soll. A ist hierbei der Startwert der Kurve, B ihre Steigung.

Ein Großprojekt mit Unity
Jagged Alliance Online

Cliffhanger Productions, die Entwickler von Jagged Alliance Online, geben Einblick in die Prozesse und Herausforderungen, die bei der Entwicklung eines großen Projekts für die Unity-Web-Player-Plattform entstehen. In Teil 1 erklärt Johannes Scharl, wie man Projekte und Daten in Unity und Visual Studio optimal organisiert.

Johannes Scharl

ist Technical Lead bei Cliffhanger Productions.

Johannes arbeitet seit 2002 als Softwareentwickler und ist seit vier Jahren in der Spielebranche tätig. Seit seinem Abschluss in Informatik an der Technischen Universität Wien ist er bei Cliffhanger Productions beschäftigt und dort aktuell als Technical Lead für die Umsetzung des Browsergames Jagged Alliance Online verantwortlich.

Jagged Alliance Online ist die moderne Neuauflage der klassischen Taktikserie. Spieler erhalten das Kommando über eine Gruppe von Soldaten und müssen in rundenbasierten Kämpfen unterschiedlichste Missionen bestehen und ihre Truppe immer weiter ausbilden. Das Spiel läuft komplett im Browser, muss also nicht installiert werden.

Für die Entwicklung von Jagged Alliance Online war die Unity Engine eine naheliegende Wahl. Dank der hohen Verbreitung des Web Players erreicht man viele Spieler, und im Gegensatz zur Flash-Plattform brauchen sich Unity-Spiele grafisch nicht hinter vollwertigen Boxed-Games zu verstecken. Sie bieten alles, was das verwöhnte Spielerauge begehrt: Echtzeitschatten, Normal Maps, physikalisch korrekte Kollisionen, Wasser- und Partikeleffekte und vieles mehr. Der Unity-Editor ist ein vollwertiger und leicht zu erweiternder Level-Editor, und Programmierern wird dank der Unterstützung von C# mit Mono die Arbeit leicht gemacht.

Bei der Entwicklung von Jagged Alliance Online stießen wir jedoch auf einige Herausforderungen, die es zu meistern gab: So machten es die umfangreiche Menge an Assets und die Größe der Codebase notwendig, das Projekt übersichtlich zu strukturieren und aufzuteilen. Ein weiteres Ziel war es, die Größe des initialen Downloads zu minimieren. Wie wir diese Herausforderungen angegangen sind, werde ich im nun folgenden ersten Teil dieser Serie erläutern.

Visual Studio Solution einrichten

Für Unity-Entwickler, die mit C# programmieren, ist Visual Studio die beste Wahl und bietet komfortable Features wie automatische Syntaxprüfung und IntelliSense. Wer davon noch nicht überzeugt ist, sollte das Tool »ReSharper« testen, das sich komplett in Visual Studio integriert und großartige Refactoring-Möglichkeiten sowie Code-Vervollständigung bietet.

Unity liefert von Haus aus sogar die automatische Generierung und Synchronisation von Visual Studio Solutions für ein Projekt aus dem Unity-Editor heraus. Dabei werden jedoch alle Skripts und Klassen in einem einzigen Namespace und Projekt angelegt. Will man seinen Code auf mehrere Libraries aufteilen, werden diese Änderungen beim nächsten Export aus Unity wieder überschrieben. Seit der Unity-Version 3.4 werden zwar automatisch mehrere Projekte in einer Solution angelegt, allerdings kann man die Aufteilung nicht selbst beeinflussen.

Ein aktueller Screenshot aus Jagged Alliance Online: Unity bietet auch im Browser alle Möglichkeiten für tolle 3D Grafik.

Gerade für größere Projekte ist es jedoch wichtig, den Programmcode in mehrere Libraries zu unterteilen (**Abbildung 1**). Tausende Klassen in einem Projekt sind unübersichtlich und führen dazu, dass die Funktionalitäten einzelner Klassen unnötig mit anderen verzahnt und deren Aufgaben unklar abgegrenzt sind. Wenn man den Code auf mehrere Projekte verteilt, hilft das, bestimmte Funktionalitäten abzukapseln. Dadurch können die Klassen einfacher getestet und wiederverwendet werden.

Ein negatives Beispiel wäre etwa eine Klasse Avatar, die folgende Funktionalitäten enthält:

- Die Logik, wie sich ein Avatar verhält und welche Aktionen er ausführen kann.
- Die Implementierung des Rendering Codes, um den Avatar in Unity zu animieren.
- Die Netzwerkfunktionalität, um Avatar-Informationen mit einem Server abzustimmen.

Stattdessen könnte man Code zum Rendern von animierten Meshes und zur Kommunikation mit einem Server in eigenen Libraries ablegen – dadurch kann diese Funktionalität leicht wiederverwendet werden.

Eine Visual Studio Solution, die mehrere Libraries enthält, ist auch für Unity einfach einzurichten. Man muss dazu nur eine eigene Solution statt der von Unity automatisch erzeugten außerhalb des Asset-Ordners anlegen (siehe Ordnerstruktur in **Abbildung 2**). Die einzelnen Projekte werden dann in Unterordnern des Asset-Ordners im Unity-Projekt angelegt und zu dieser Solution hinzugefügt. Dadurch kann man einfach neue Dateien in Visual Studio anlegen, umbenennen, löschen, den Code kompilieren und testen. Unity verwendet dabei automatisch die gespeicherten C#-Dateien. Auf diese Weise können sogar Namespaces in den Klassen verwendet werden. Auch Live-Debugging ist mit dieser selbst angelegten Solution möglich, allerdings nur über die von Unity angepasste »MonoDevelop IDE«. Mit dieser lässt sich die Solution öffnen und man kann sich an einen laufenden Unity-Prozess anhängen. Danach kann man Variablen inspizieren und Breakpoints setzen.

Code, der nicht direkt von Unity benötigt wird (z.B. Unit- oder Integration-Tests), kann man außerhalb des Asset-Ordners anlegen und trotzdem als eigenes Projekt zur Solution hinzufügen und kompilieren.

Eine solche Custom-Solution muss allerdings noch leicht angepasst werden, damit sie problemlos mit Unity verwendet werden kann. Für jede in Visual Studio angelegte Library wird automatisch eine Datei »AssemblyInfo.cs« erzeugt, die Meta-Daten wie Versionsnummer, Konfiguration oder Entwickler enthält. Weil diese Datei für jedes Projekt den gleichen Namen hat, führt das zu Konflikten in Unity. Um dies zu vermeiden, kann man die AssemblyInfo.cs einfach löschen, weil sie in dieser Konfiguration nicht weiter gebraucht wird.

Ein anderes Problem ist, dass der Code einer Library von Visual Studio per Default als Dynamic Link Library (DLL) in ein Unterverzeichnis

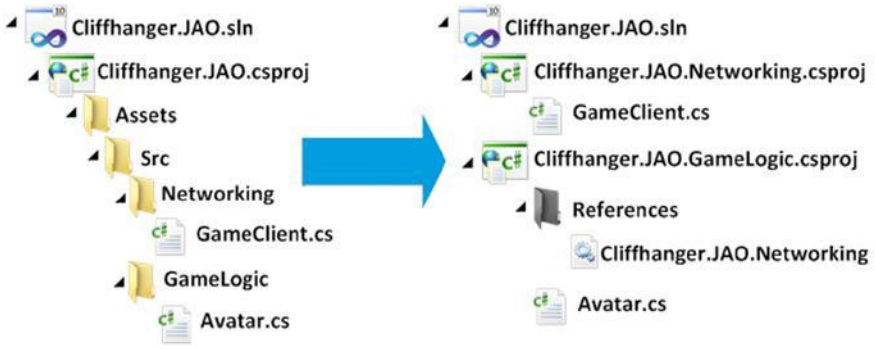


Abbildung 1: Statt alle Klassen in einem einzigen Projekt anzulegen, sollte die Funktionalität in verschiedenen Libraries abgekapselt sein.

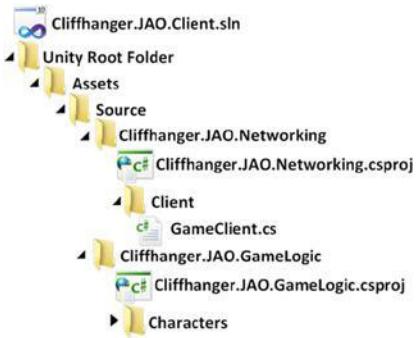


Abbildung 2: In einer **selbst erstellten Visual Studio Solution** kann man Source Code leicht in verschiedene Libraries aufteilen. Bei der automatisch aus Unity synchronisierten Solution ist dies nicht möglich.

```
<PropertyGroup Condition=" '$(Configuration)|$(Platform)' == 'Debug|AnyCPU' ">
  <BaseIntermediateOutputPath>..\..\..\Temp\obj\Debug\</BaseIntermediateOutputPath>
  <OutputPath>..\..\..\Temp\bin\Debug\</OutputPath>
</PropertyGroup>
<PropertyGroup Condition=" '$(Configuration)|$(Platform)' == 'Release|AnyCPU' ">
  <BaseIntermediateOutputPath>..\..\..\Temp\obj\Release\</BaseIntermediateOutputPath>
  <OutputPath>..\..\..\Temp\bin\Release\</OutputPath>
</PropertyGroup>
```

Abbildung 3: Die Pfade für BaseIntermediateOutputPath und OutputPath einer ».csproj«-Datei müssen manuell angepasst werden, um zu verhindern, dass **Assemblies** direkt in die Unity-Projektstruktur kompiliert werden.

kompiliert wird. Diese Datei kann in Unity auch direkt statt den C#-Klassen verwendet werden. Wenn allerdings beide im Unity-Asset-Ordner liegen, kommt es wieder zu Konflikten. Deswegen muss man die Assemblies in ein Verzeichnis außerhalb der Asset-Struktur kompilieren, zum Beispiel in den von Unity angelegten Temp-Ordner. Am einfachsten erreicht man dies, indem man einmalig die Settings für den »BaseIntermediateOutputPath« und den »OutputPath« jeder Konfiguration in der ».csproj«-Datei einer Library manuell in einem Texteditor anpasst (siehe Code in **Abbildung 3**).

Code gemeinsam nutzen

Jagged Alliance Online verwendet eine klassische Client-Server-Architektur: Der Unity Web Player Client verbindet sich zu einem Game-Server, der in C# entwickelt ist und als .NET-Applikation auf einem Windows-Server läuft. Dieser Server kontrolliert die Spielmechanik, speichert den Fortschritt, synchronisiert Multiplayer-Spiele und bestimmt darüber hinaus das Verhalten der Computergegner.

Abbildung 4: Die **Logik zur Berechnung der** Treffergenauigkeit wird sowohl auf dem Unity Client als auch auf dem Game-Server verwendet.

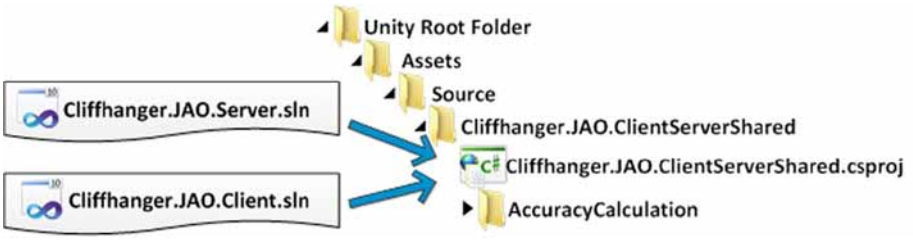


Abbildung 5: Code, der einerseits vom Unity Client und andererseits vom Game-Server verwendet wird, kann in eine **eigene Library** ausgelagert und von mehreren Solutions referenziert werden.

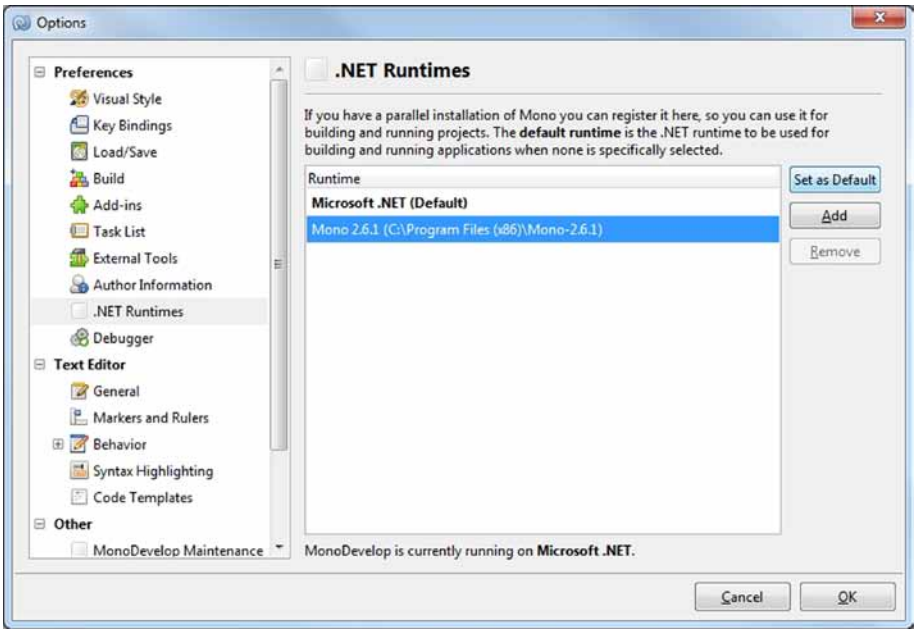


Abbildung 6: Um in Unity DLLs debuggen zu können, muss man die richtige **Mono Runtime** konfigurieren.

Große Teile der Funktionalität unterscheiden sich kaum zwischen Client und Server. Ein Beispiel dafür ist die Berechnung der Treffergenauigkeit. Diese wird im Unity Client angezeigt, wenn man mit einem Söldner auf einen Gegner zielt. Sie ergibt sich aus den Attributen beider Einheiten, deren Entfernung und den Werten der verwendeten Waffe. Genau derselbe Wert wird auf dem Game-Server für die Berechnung des Trefferschadens benötigt (**Abbildung 4**).

Um die Kalkulation der Treffergenauigkeit nicht zwei Mal implementieren zu müssen, sollte derselbe Code sowohl auf dem Client als auch auf dem Server verwendet werden – weil wir beides mit C# entwickeln, ist das auch nicht weiter schwierig. Für den gemeinsam verwendeten Code kann man ein oder mehrere Visual-Studio-Projekte zu der zuvor angelegten Solution hinzufügen (**Abbildung 5**). Diese können dann auch von einer Server-Solution referenziert werden, oder man kann gar eine gemeinsame Solution für Server und Client-Code anlegen. Wichtig ist nur, darauf zu achten, dass das Target-Framework dieser Libraries auf .NET 3.5 eingestellt ist. Damit ist der Code kompatibel zu der aktuell in Unity verwendeten Mono-Version und kann trotzdem auch in .NET 4.0-Applikationen verwendet werden.

Unity-Code ausschließlich per DLLs bereitstellen

Eine andere Möglichkeit, eine größere Codebase für Unity zu organisieren, hat die Firma Schell Games auf der Unite 11 präsentiert. Statt die Visual-Studio-Projekte inklusive aller Source-Code-Dateien im Asset-Ordner des Unity-Projekts zu speichern, kann man auch den gesamten Code ausschließlich als kompilierte DLLs in Unity bereitstellen. Dadurch kann Funktionalität noch einfacher wiederverwendet werden – entweder in weiteren Unity-Projekten, oder für in .NET entwickelten Game-Servern. Wenn man diesen Weg einschlagen möchte, gilt es aber, die folgenden wichtigen Punkte zu berücksichtigen:

Von Haus aus ist das Debuggen von Code in DLLs mit MonoDevelop nicht ohne Weiteres

möglich – dazu muss man erst einige spezielle Einstellungen vornehmen. Zunächst benötigt man die von MonoDevelop erzeugten ».mdb«-Dateien. Sie enthalten die Debug-Informationen zu einer DLL und müssen neben dieser im Unity-Projekt abgelegt werden. Unter OSX ist MonoDevelop bereits fertig dafür konfiguriert, aber Windows-Benutzer müssen die richtige Konfiguration noch selbst vornehmen. Wenn man das zu Unity kompatible Mono-SDK von der offiziellen Website (mono-project.com/OldReleases) installiert, kann man in MonoDevelop unter »Tools« und »Preferences« das gerade installierte Mono-Framework als Default .NET Runtime einstellen (**Abbildung 6**).

Mit diesen Anpassungen kann man MonoDevelop wie gewohnt mittels »Run« und »Attach To Process« mit einem laufenden Unity-Prozess verbinden und die Applikation debuggen. Auch Breakpoints und das Inspizieren von Variablen funktioniert normal.

In einem größeren Team wird meist mit einer Versionsverwaltung wie SVN oder Perforce gearbeitet. Bei solchen Systemen wird im Allgemeinen nur der Source-Code versioniert, weil parallele Änderungen von mehreren Personen an textbasierten Dateien einfach zusammengeführt werden können. Bei binären Daten wie .NET-Assemblies ist das nicht möglich. Aus diesem Grund kommt es zu Konflikten, wenn zwei Personen gleichzeitig etwas an einer Library verändert haben und diese in die Versionsverwaltung einchecken wollen. Wenn Code für Unity ausschließlich mit DLLs bereitgestellt wird, kommt es bei einem größeren Team wesentlich öfter zu diesen Konflikten, weshalb die einzelnen Entwickler diese Dateien nicht einchecken sollten. Stattdessen sollte dafür ein automatisierter Prozess eingerichtet werden, der in gewissen Intervallen den Code kompiliert und in die Versionsverwaltung eincheckt. Entwickelter Code kann so lokal kompiliert und getestet werden, ohne dass ständig Repository-Konflikte gelöst werden müssen.

Kurioses Detail am Rande: Weil Unity direkt den Common Intermediate Language Code der .NET-Assemblies mit Mono ausführt, muss man nicht zwangsweise C# verwenden. Wer möchte, kann auf jede .NET-kompatible Sprache zur Entwicklung seines Unity-Spiels zurückgreifen, wie etwa IronPython oder F#.

Ein Unity-Projekt auf mehrere Projekte aufteilen

Für Jagged Alliance Online hatten wir von Anfang an hohe Ansprüche an den Content. So wird das Spiel zum Release an die hundert Levels bieten, die in unterschiedlichsten Settings angesiedelt sind – von Militärcamps in der Wüste über idyllische Strände bis hin zu urbanen Nachtszenarien und Villen von Drogenbaronen. Dazu kommen rund 60 unterschiedliche Söldner und Dutzende Gegnertypen.

Diese Menge an Content bringt allerdings auch Probleme mit sich. Weil von unseren Artists und Level-Designern Hunderte Assets pro Tag hinzugefügt und geändert werden, muss auch der Rest des Teams regelmäßig all diese Texturen und Models in Unity importieren. Zeitweise ging über eine Stunde pro Tag und Person für das Aktualisieren und Importieren von Content in Unity verloren. Hier einige Zahlen: Aktuell umfasst unsere Repository-Datenbank knapp 15.000 Revisionen und fast 100 Gigabyte mit über 150.000 versionierten Daten.

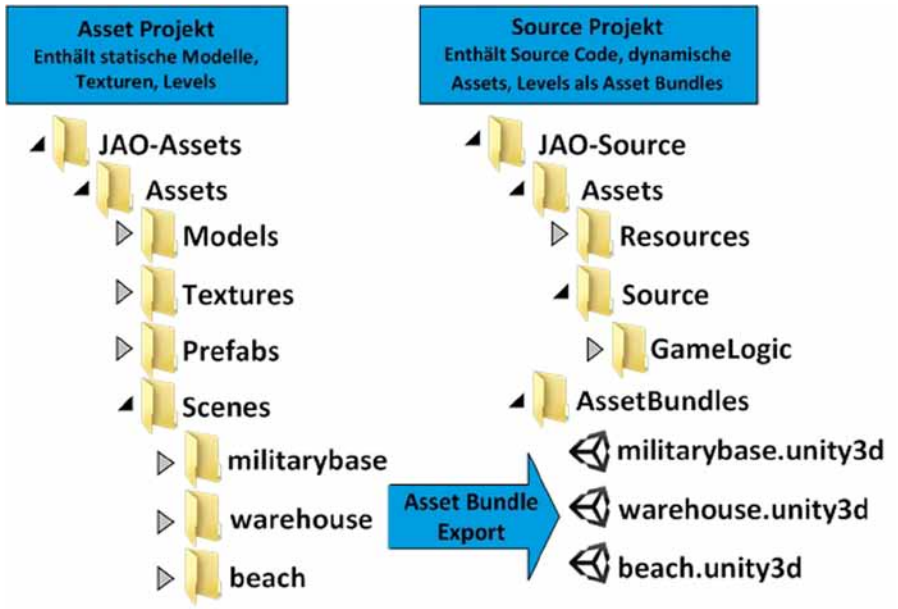
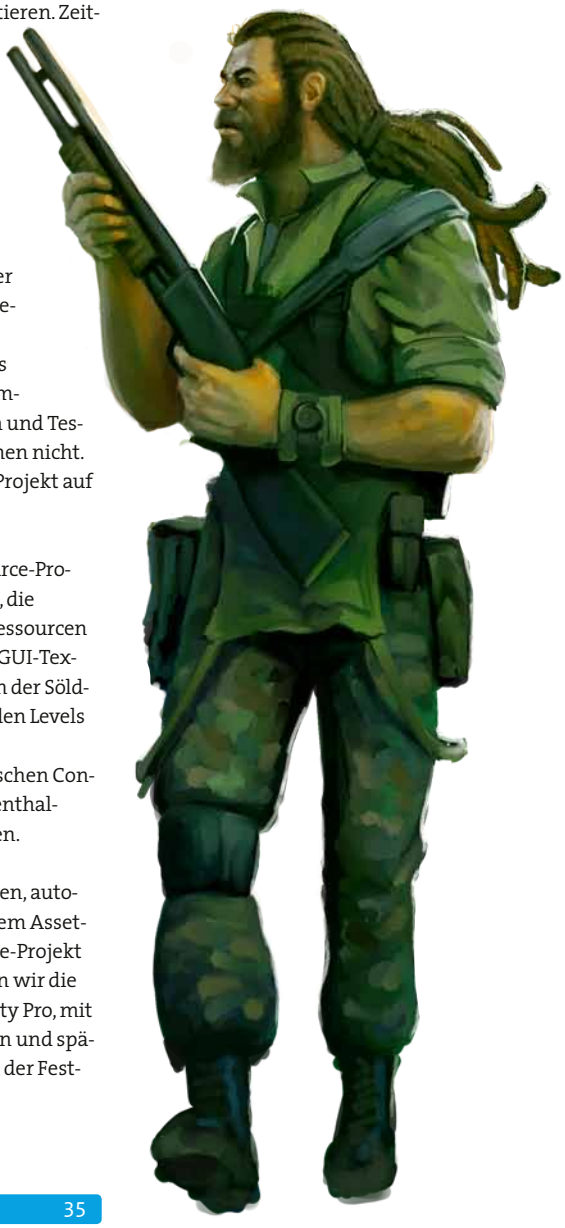


Abbildung 7: Der statische Content für Jagged Alliance Online befindet sich in einem eigenen Unity-Projekt. Die Levels werden als **Asset Bundles** exportiert und aus dem Source-Projekt heraus geladen.

Dabei sind vor allem Programmierer bei ihrer Arbeit nur selten darauf angewiesen, immer die letzten Assets verwenden zu können – fehlende Models in Levels, falsche Light Maps oder Dummy-Texturen stören beim Entwickeln und Testen von neuen Features im Allgemeinen nicht. Aus diesem Grund wurde das Unity-Projekt auf zwei Projekte aufgeteilt:

- Die Programmierer arbeiten im Source-Projekt, das den gesamten Source Code, die Startszene und alle dynamischen Ressourcen enthält. Darunter fallen neben den GUI-Texturen auch die Models und Texturen der Söldner und Gegner, die zur Laufzeit in den Levels instanziiert werden.
- Das Asset-Projekt enthält den statischen Content, also alle Levels mit den darin enthaltenen Terrains, Models und Texturen.

Fertige Levels werden in einem eigenen, automatisierten Schritt regelmäßig aus dem Asset-Projekt exportiert und aus dem Source-Projekt zur Laufzeit geladen. Dazu verwenden wir die Asset-Bundle-Funktionalität von Unity Pro, mit der man Content als Paket exportieren und später sowohl aus dem Web als auch von der Festplatte laden kann (**Abbildung 7**).



Cliffhanger Productions

Cliffhanger Productions ist ein unabhängiges Produktions- und Entwicklungsstudio für Online- und Rollenspiele mit Sitz in Wien und Frankfurt. Seit der Gründung 2005 war Cliffhanger Productions an vielen erfolgreichen deutschsprachigen Produktionen beteiligt, wie etwa Gothic 3, Risen oder Spellforce 2.
Im internen Entwicklungsstudio beschäftigt sich ein Team von erfahrenen Programmierern, Designern und Artists mit der Entwicklung der Browsergames Jagged Alliance Online und Shadowrun Online. Das Cyperpunk-Rollenspiel ist der erste eigene Titel von Cliffhanger Productions.

Die Aufteilung in mehrere Unity-Projekte hat folgende Vorteile:

- Es müssen nicht mehr alle Assets regelmäßig von allen Entwicklern in einem zeitintensiven Prozess importiert werden, weil der Großteil des Contents als gepackte Asset Bundles erst zur Laufzeit geladen wird.
- Die Download-Größe der Web-Player-Datei, mit der das Spiel gestartet werden kann, wird merklich reduziert, weil keine Levels mehr enthalten sind. Sie werden einzeln geladen, wenn der Spieler eine Mission startet.
- Durch die Trennung von Code und Content erreicht man außerdem eine deutliche Entkopplung der Logik von den Levels. Gerade bei größeren Projekten ist diese Trennung wichtig, um leichter neue Levels hinzufügen und den Code wiederverwenden zu können. Ist die Logik zu sehr an die Objekthierarchie eines Levels gebunden, wird die Erweiterung schwierig.

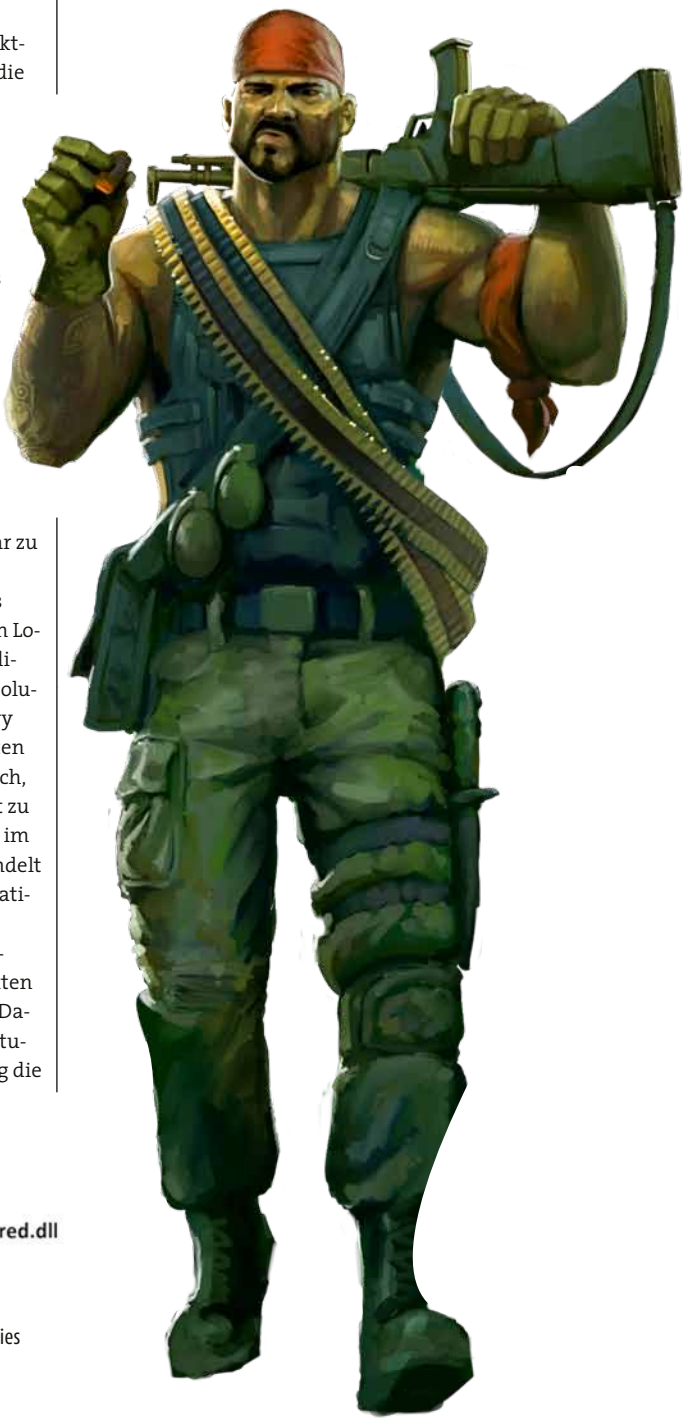
In unserem Fall ist nur ein sehr kleiner Teil des Codes im Asset-Projekt verfügbar – nämlich nur die Skripte, die von den Level-Designern in den Szenen platziert werden müssen. Bei Jagged Alliance Online sind das etwa die Pathfinding-Komponente für die Charaktere, interaktive Objekte (wie zerstörbare Wände) oder Trigger Boxes, die ein bestimmtes Verhalten der KI auslösen.

Der Code, der sowohl vom Source- als auch vom Asset-Projekt verwendet wird, sollte zwar minimal gehalten werden, es benötigt jedoch einen kleinen Trick, um ihn in beiden Projekten verfügbar zu machen. Schließlich müssen die gleichen Skripte, die als Komponenten in den Levels verwendet werden, auch von der restlichen Logik referenziert werden, ohne Code zu duplizieren. Mit der zuvor angelegten Custom-Solution ist es ein Leichtes, eine separate Library anzulegen, die den gemeinsam verwendeten Code enthält. Leider ist es aber nicht möglich, den Source Code einfach im Source-Projekt zu belassen und die kompilierten Assemblies im Asset-Projekt zu verwenden. Für Unity handelt es sich dann um unterschiedliche, inkompatible Klassen. Die beste Lösung ist daher, die Source-Code-Dateien außerhalb der Unity-Projekte zu speichern und in beiden Projekten die erzeugten DLL-Dateien zu verwenden. Dazu kann man Post-Build-Events in Visual Studio einrichten, die bei jedem Build-Vorgang die Assemblies automatisch in die Asset-Ordner beider Projekte kopieren (**Abbildung 8**).

Abschließend bleibt zu sagen, dass Unity trotz der besprochenen Herausforderungen auch für größere Projekte eine sehr gute Wahl ist. Gerade für Browsergames oder Spiele für mobile Plattformen bietet die Engine umfangreiche Features und ermöglicht tolle Ergebnisse in kurzer Zeit.

Im zweiten Teil dieser Serie wird erklärt, wie Unit-Tests für Unity implementiert werden können. Außerdem wird erläutert, wie das vollautomatische Deployment von Jagged Alliance Online funktioniert und welche Herausforderungen zu meistern waren, um dessen Performance und Stabilität zu optimieren.

Johannes Scharl



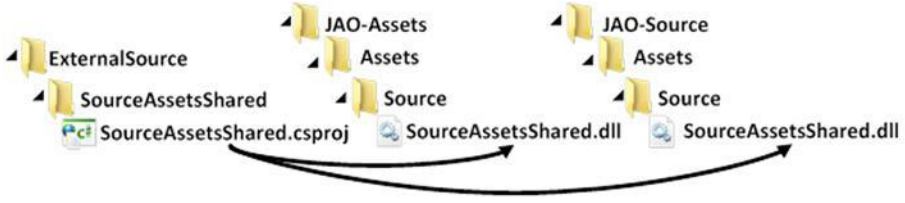


Abbildung 8: Um Source Code in mehreren Unity-Projekten zu verwenden, kann man kompilierte Assemblies einfach in beide **Asset-Ordner** kopieren.

making games TALENTS

DIE ERFOLGREICHSTE TALENTBÖRSE FÜR SPIELE-ENTWICKLER

DEIN WEG IN DIE SPIELEBRANCHE!

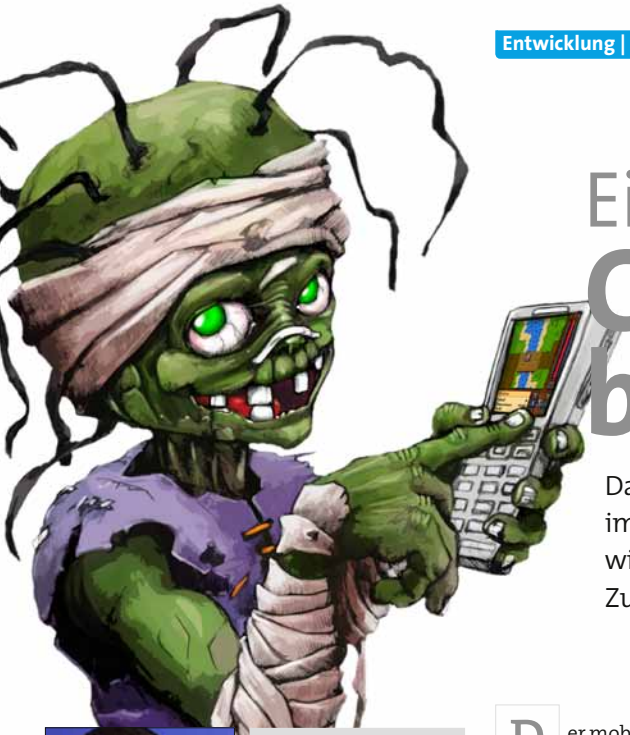
Mehr als 200 Stellen haben wir schon vermittelt. Ergreif jetzt deine Chance auf einen Traumjob!

28. Januar 2012

IDG Medienhaus München

Anmeldeinformationen unter
www.makinggames.de/talents

unter anderem mit: TRAVIAN GAMES, CRYTEK, CipSoft online entertainment, CHIMERA Entertainment, Trollgames



Ein Spiel, Hunderte Geräte
Cross-Plattform bei TibiaME

Das Rollenspiel TibiaME hält sich bereits seit acht Jahren erfolgreich im Markt. Dabei musste CipSoft das Multiplattform-MMO immer wieder den neuen Smartphone-Technologien anpassen. Benjamin Zuckerer erklärt, wie die Regensburger dies geschafft haben.

Benjamin Zuckerer

ist Produktmanager bei CipSoft.

Benjamin ist seit 2007 für die Entwicklung des weltweit ersten mobilen MMOGs TibiaME verantwortlich. Er kümmerte sich dabei um die Planung, das Design, den Betrieb und die Entwicklungsphasen des Spiels.

Der mobile Markt verändert sich rasend schnell. Noch vor einigen Jahren war Nokia der erfolgreichste Hersteller von Mobiltelefonen und hatte Millionen Geräte mit dem Betriebssystem Symbian OS am Markt verkauft. Dann kamen Apple mit seinem iPhone und Google mit dem offenen Betriebssystem Android und alles wurde gehörig durcheinandergewirbelt. Wie kann man als Entwickler auf diese Veränderungen reagieren und die wichtigsten Smartphone-Plattformen abdecken?

Die Ausgangslage

CipSoft hat 2003 mit TibiaME das weltweit erste MMOG für Mobiltelefone veröffentlicht. Damals lief das Spiel auf Symbian- und J2ME-Geräten. Diese hatten üblicherweise ein kleines Display (128x128 Pixel oder 176x208 Pixel) und zusätzlich zum Ziffernblock noch einen kleinen Joystick. Die Eingabemöglichkeiten waren also sehr begrenzt und es gab nur wenig Platz, um das eigentliche Spiel darzustellen. Entsprechend wurden die Spielgrafiken in 20x20 Pixel (siehe **Abbildung 1**) gezeichnet und kamen mit nur wenigen Animationsphasen aus. Sound und Musik gab es aus Performancegründen überhaupt nicht. Die Steuerung wurde für Mobiltelefone optimiert, und es wurde viel Wert auf ein schlankes Protokoll gelegt (Internet-Flatrates waren noch eine absolute Seltenheit).

Als sich die schönen neuen Smartphones mit großen Touchdisplays und ständiger Konnektivität am Markt durchsetzten, mussten wir feststellen, dass unser »Classic«-Client auf diesen Geräten einfach nicht mehr funktionierte. Das vollkommen andere Bedienkonzept (Berührung des Displays statt Tastatur und Joystick) und die hochauflösenden Bildschirme erforderten eine komplette Neuentwicklung. Zu diesem Zeitpunkt wollten wir die neuen »Symbian Series 60 v5«-Geräte (allesamt mit Touchdisplay) und die iOS-Plattform unterstützen. Um nicht für jede Plattform einen eigenen Client neu schreiben zu müssen, entschieden wir uns für einen Cross-Plattform-Ansatz. Inzwischen ist TibiaME zusätzlich noch für Android-Smartphones verfügbar und kann auch direkt im Browser mittels Java-Applet gespielt werden.

Abbildung 1: TibiaME auf einem **Symbian-Mobiltelefon** mit noch sehr geringer Auflösung.

Warum Cross-Plattform entwickeln?

Es gibt viele Gründe, ein Spiel auf mehreren Plattformen parallel zu veröffentlichen, gerade im mobilen Sektor. Durch den großen Erfolg der App Stores tummeln sich sehr viele Entwickler im Markt und wollen ihre Spiele an den Kunden bringen. Der Konkurrenzdruck ist dadurch sehr groß. Durch die Cross-Plattform-Entwicklung kann CipSoft das daraus resultierende Risiko auf mehrere Plattformen verteilen. Zudem können wir schneller auf Marktveränderungen reagieren. Wenn eine neue Smartphone-Plattform startet (z.B. Blackberry QNX), können wir in kurzer Zeit einen lauffähigen Client fertigstellen und somit von Anfang an im neuen Store präsent sein – ein klarer Wettbewerbsvorteil! Neue Features und Bugfixes müssen nur einmal programmiert werden und stehen dann für alle Plattformen zur Verfügung. Wir sparen uns dabei also Zeit und können allen Kunden das gleiche Spielerlebnis bieten. Jede zusätzliche Plattform bedeutet gleichzeitig auch mehr potenzielle Kunden und bringt einen stetigen Fluss neuer Spieler (App-Entdeckung). Ein weiterer Aspekt ist, dass die User gemeinsam mit ihren Freunden spielen können, egal für welches Mobiltelefon sie sich jeweils entschieden haben. Gerade bei einem Multiplayer-Titel wie TibiaME ist das natürlich wünschenswert.

Architektur

Weil viele der neuen Smartphone-Plattformen Native Code unterstützen, haben wir uns dazu entschlossen, den Großteil des Codes in C++ zu schreiben. Die Architektur des neuen Clients (Extreme Client) wird dafür in mehrere Schichten aufgeteilt (siehe **Abbildung 2**).

Die unterste Schicht (der Core) ist dabei für die Datenverarbeitung, die Netzwerkkommunikation sowie das Layout von Grafiken und User Interface (UI) zuständig. Der gesamte Core ist in C++ geschrieben. Die nächste Schicht kümmert sich um die Grafikausgabe. Hierfür wird OpenGL verwendet, der Code ist in C geschrieben. Weil allerdings nicht alle Plattformen OpenGL voll unterstützen (z.B. Symbian), gibt es hier auch native Renderer. Der Code dieser zwei Ebenen macht zusammen ungefähr 80 Prozent des gesamten Codes aus. Auf der obersten Schicht befindet sich der plattformspezifische Teil. Dieser ist für die Eingabe (Touch Events, Accelerometer etc.), Systemdialoge und OS-spezifische Funktionen zuständig. Bei iOS wird mit C++ und Objective C gearbeitet. Symbian hat seinen eigenen C++-Dialekt. Für den in Dalvik geschriebenen Android- und den in Java geschriebenen Web-Client ist allerdings das Java Native Interface noch als Zwischenschicht notwendig, um auf die darunterliegenden C++-Schichten zugreifen zu können. Der plattformspezifische Teil veranschlagt in etwa 20 Prozent (pro Plattform).

Entwicklungsumgebung und Tools

Für den iOS-Client benötigt man zwangsweise einen Mac-Rechner und außerdem die Entwicklungsumgebung XCode. Die restliche Programmierung findet in Eclipse mit den entsprechenden Plugins statt (zum Beispiel Android NDK). Wir verwenden darüber hinaus noch einige externe Bibliotheken: OpenGL ES 1.1 für das Rendering, OpenAL für die Wiedergabe von Sound und Musik, Libpng zum Laden von Grafikassets und RapidXML zum Parsen der Interface-Beschreibungsdateien (hierzu noch später mehr). Um die Grafiken automatisch auf unterschiedliche Größen zu skalieren, verwenden wir außerdem die Software ImageMagick.

Lokalisierung

TibiaME ist weltweit verfügbar und unterstützt mehrere Sprachen. Besonders erfolgreich ist das Spiel im asiatischen Raum, zum Beispiel in Indonesien oder Brunei. Unsere Spieler können das gesamte Spiel in Deutsch, Englisch und Malaiisch spielen. Dabei werden neben den eigentlichen Spieltexten (NPCs, Objekten, Beschreibungen etc.), die auf unseren Servern liegen, auch alle Texte in den Clients (Menüs, Fehlermeldungen etc.) übersetzt. Der Extreme Client muss also mehrere Sprachen verwalten und darstellen können. Dafür verwenden wir das offene Format XLIFF. Alle benötigten Sprachlabels werden in entsprechenden Sprachdateien gespeichert. Für die eigentliche Übersetzung verwenden wir ein tolles Open-Source-Tool, den Pootle Server. Pootle ermöglicht es dem Lokalisierungsstudio, alle Texte über eine Browseranwendung zu übersetzen. Als Entwickler haben wir so jederzeit Einblick in den Fortschritt des Lokalisierungsstudios und können die fertig übersetzten Dateien ohne zusätzliche Konvertierung direkt im Client oder Server verwenden.

Vorüberlegungen

Am Anfang der Cross-Plattform-Entwicklung sollte man sich folgende Fragen stellen:

1. Welche Plattformen sollen von dem Spiel unterstützt werden?
Man sollte vor allem langfristig planen und bereits zukünftig interessante Plattformen in den Überlegungen berücksichtigen.

2. Setzt das Spiel eine bestimmte Hardware für den flüssigen Ablauf voraus?
Hier spielen vor allem CPU, Speicherbedarf und die benötigte Grafikpower eine Rolle.

3. Sind für das Spiel bestimmte Features eines Betriebssystems notwendig?
Um etwa In-App Payment im Client anbieten zu können, muss man bei iOS und Android eine bestimmte OS-Version installiert haben. Bei Android muss auch noch die Market App in einer bestimmten Version installiert sein.

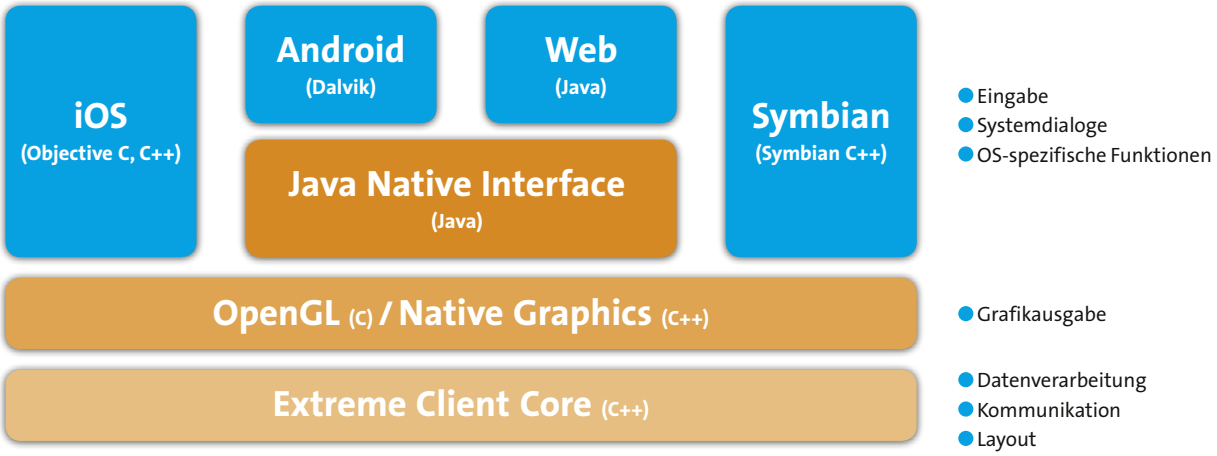


Abbildung 2: Das **Schichtenmodell** des TibiaME Extreme Clients.

Layout 1: Seitenverhältnis: 3:4 (240x320 Pixel)

Layout 2: Seitenverhältnis: 2:3 (320x480 Pixel)

Layout 3: Seitenverhältnis: 9:16 (360x640 Pixel)

Damit man nicht von schwächeren Plattformen oder Mobiltelefonen eingeschränkt wird, sollte man so viele Features wie möglich optional gestalten. Das Spiel kann dadurch alle Features der neuen Geräte nutzen, dabei aber trotzdem noch auf den älteren Mobiltelefonen laufen, auch wenn es hier eventuell nicht mehr ganz so schön aussieht. Wichtig ist allerdings, alle Annahmen über die Leistungsfähigkeit immer auch mit einem Prototyp zu testen.

Technische Herausforderungen

Eine große Herausforderung ist die Fragmentierung. Auf der einen Seite entsteht Fragmentierung durch die Unterstützung weiterer Plattformen, auf der anderen Seite gibt es auch innerhalb der Plattformen Fragmentierung. Besonders bei Android fällt dies auf. Es gibt Hunderte unterschiedliche Android-Mobiltelefone. Diese basieren auf teils sehr unterschiedlicher Hardware (CPU-Geschwindigkeit, GPU-Geschwindigkeit, verfügbarer Speicher), sowie unterschiedlichen OS-Versionen, und manchmal sind vom Hersteller noch weitere Änderungen vorgenommen worden. Über alle Plattformen hinweg stellen die vielen andersartigen Bildschirmauflösungen das größte Problem dar – sie reichen von QVGA (240x320 Pixel) bis hin zum Retina-Display (640x960 Pixel). Zudem haben die Bildschirme auch noch ungleiche Seitenverhältnisse. Weil wir für TibiaME pixelbasierte Grafiken verwenden, können wir diese nicht ohne Weiteres auf den Geräten skalieren. In den meisten Fällen reicht die Performance nicht aus, um die Grafiken ohne massiven Qualitätsverlust zu skalieren. 3D-Spiele haben es an dieser Stelle deutlich leichter. Wir können auch nicht das gleiche UI-Layout für alle Geräte verwenden, weil die Unterschiede einfach zu groß sind. Die Handyhersteller bringen regelmäßig neue Telefone heraus und lassen sich dabei auch immer wieder etwas Neues einfallen (Auflösung, Pixeldichte, Seitenverhältnisse). Bei TibiaME haben wir uns für drei Maßnahmen entschieden, die es uns erlauben, auch zukünftige Geräte zu unterstützen.

Maßnahme 1: vorab erstellte Assets

Alle Assets werden bereits vorab passend erstellt. Die Grafiken werden in 64x64 Pixeln gezeichnet und anschließend mithilfe von ImageMagick auf alle möglichen Größen zwischen 35x35 Pixeln und 99x99 Pixeln skaliert. Wir stellen diese Grafiken dann auf unseren Servern zum Download bereit. Nach dem ersten Start des Clients wird bestimmt, welche Grafikassets heruntergeladen werden müssen, um den Bildschirm optimal auszufüllen.

Maßnahme 2: dynamisches UI

Das Gleiche gilt für das UI-Layout. Anhand von verschiedener Kriterien (Bildschirmauflösung, Seitenverhältnis, Pixeldichte etc.) wird errechnet, welches Layout am besten für das Gerät geeignet ist. Für die UI-Layouts verwenden wir eine mächtige XML-Beschreibungsdatei. Diese erlaubt es uns, mithilfe diverser Regeln die Position der UI-Elemente für unterschiedliche Seitenverhältnisse und auch Betriebssysteme festzulegen. So können wir zum Beispiel bestimmte UI-Elemente deaktivieren (also nicht anzeigen) oder je nach Bildschirmausrichtung (Portrait oder Landscape) anzeigen. Soll auf Symbian der Scrollbalken dicker sein (damit er auch mit einem Stylus verwendet werden kann), so lässt sich diese Unterscheidung mit den Filtern regeln. Die Filter berücksichtigen aber auch die vorhandenen Eingabemöglichkeiten des Geräts, also ob es einen Touchscreen, ein Keyboard oder beides zur Verfügung hat. Um Platz zu sparen, verwenden wir – falls möglich – 9-Grid-Grafiken. Hierbei wird die Grafik in vier Eckteile und vier Seitenteile geordnet. Die Seitenteile können dann beliebig oft aneinandergefügt werden, womit das UI-Element ohne Probleme skaliert werden kann (z.B. Inventarelement). Ist dies nicht möglich (z.B. beim virtuellen Joystick), bieten wir das UI-Element in unterschiedlichen Größen an: Orginal, 125 Prozent, 150 Prozent und 200 Prozent. Aktuell verwenden wir drei unterschiedliche UI-Layouts mit jeweils einer Portrait- und einer Landscape-Variante, welche die häufigsten Bildschirmauflösungen abdecken (siehe **Layouts 1 bis 3**).

Auf den Screenshots kann man sehr gut erkennen, wie das dynamische UI funktioniert. Bei Layout 3 und Layout 2 werden dieselben UI-Elemente verwendet. Um den unterschiedlichen Seitenverhältnissen gerecht zu werden, erfolgt die Anordnung in leicht abgeänderter Form. So wird die Menüleiste (inklusive der aktiven Waffe) in Layout 3 über dem Inventar gezeichnet. Bei Layout 2 befindet sie sich am rechten Bildschirmrand. Auch die Minimap weicht ab, in Layout 3 wird ihr doppelt so viel Platz eingeräumt und dieser optimal ausgenutzt. Bei Layout 1 reicht der Platz aufgrund der geringen Bildschirmauflösung nicht aus. Hier müssen also einige UI-Elemente weggelassen und die Anordnung grundlegend verändert werden. Der virtuelle Joystick wird nicht angezeigt, das Inventar und die aktive Waffe sind auf zwei Zeilen zusammengestaucht und die Minimap am rechten Bildschirmrand angeordnet. Bei Mobiltelefonen mit Touchdisplay und Layout 1 ändert sich das UI nochmals. Die Minimap wird dann als Overlay-Element im Spielbereich angezeigt, um Platz für eine Menüleiste zu schaffen.

An Layout 2 lässt sich auch schön erklären, wie UI-Elemente in unterschiedlichen Größen (Orginal, 100 Prozent, 150 Prozent und 200 Prozent) verwendet werden. Das Orginal-iPhone hat einen Bildschirm mit einer Auflösung von 320x480 Pixeln verbaut. Das iPhone 4 und das iPhone 4S besitzen bereits ein Retina-Display mit der doppelten Auflösung (640x960 Pixeln). Beide Bildschirme haben dasselbe Seitenverhältnis 2:3. Auf den Retina-Displays werden einfach die 200-Prozent-Grafiken verwendet, womit das Bild gestochen scharf wirkt. Das Layout bleibt bei beiden vollkommen identisch.

Maßnahme 3: Steuerung des Charakters

Bei der Cross-Plattform-Entwicklung sollte man bedenken, dass die Geräte teilweise sehr unterschiedliche Eingabemöglichkeiten haben. So gibt es Telefone mit Keyboard, Touchscreen oder gar beiden Eingabemethoden. Dabei funktionieren nicht alle Touchdisplays gleich. Es gibt Telefone (egal welche Technik dahinter steht), bei denen die Eingaben einfach flüssiger von der Hand gehen. Weil TibiaME auf Hunderten von Geräten laufen soll, muss das bei der Bedienung berücksichtigt werden. Wir haben deshalb absichtlich auf ausgefallene Gesten (z.B. Mehrfingergesten, Drag & Drop etc.) verzichtet. Stattdessen haben wir eine Auswahl an Steuerungsmöglichkeiten implementiert. Der Spieler kann sich dann selbst die für ihn angenehmste aussuchen. Am offensichtlichsten ist das einfache Tippen auf den Bildschirm, das den Charakter an die entsprechende Stelle laufen lässt. Eine weitere Steuerung wird über den virtuellen Joystick zur Verfügung gestellt. Viele Spieler erkennen auf den ersten Blick, wie dieser funktioniert. Für die fortgeschrittenen Spieler gibt es auch noch die Möglichkeit, den Charakter mit Wischbewegungen (Swipe) zu steuern.

Fazit

Mit unserem Cross-Plattform-Ansatz konnten wir TibiaME auf den Plattformen Symbian Series 60 v3, Symbian Series 60 v5, iOS, Android und im Web als Java-Applet veröffentlichen. Dabei haben zwei Programmierer etwa eineinhalb Jahre am Core und an der iOS- und Symbian-Implementierung gearbeitet. Android und die beschriebenen Maßnahmen bezüglich Fragmentierung haben einen Programmierer weitere vier Monate beschäftigt. Für den Web-Client haben wir hingegen nur noch einen Monat Programmierzeit (ein Programmierer) benötigt. Mit dem Extreme Client hat CipSoft jetzt ein solides Grundgerüst, mit dessen Hilfe wir neue Plattformen schnell abdecken können.

Benjamin Zuckerer

Die **upjers GmbH & Co. KG** ist Entwickler und Publisher von Browserspielen und zählt zu den führenden Herstellern der Browsergame-Branche.

Head of Online-Marketing (m/w)

Deine Aufgaben:
- Du koordinierst und leitest das Marketingteam sowie andere Abteilungen (u.a. Business-Development, Internationalisierung)
- Du bist für die Planung und Durchführung performaceorientierter SEM-Kampagnen zuständig
- Du kümmerst Dich um das Controlling, die Analyse und das Reporting der Marketingmaßnahmen
- Du bist für die Verfolgung von Marketingtrends und deren Erfolgskontrolle zuständig

Dein Profil:
- Du verfügst über ein abgeschlossenes Studium, idealerweise mit Schwerpunkt Marketing, Vetrieb oder E-Business
- Du besitzt praktische Erfahrungen im Online-Marketing (zwingend erforderlich)
- Du beherrscht verhandlungssicher Englisch
- Du bist eine Führungspersönlichkeit mit entsprechenden Erfahrungen
- Du besitzt fundierte Kenntnisse des Browsergame-Marktes

weitere Infos auf: **jobs.upjers.com**

Vorsicht ist besser als Nachsicht

10 Pre-Production-Irrtümer

Kaum eine Projektphase ist so entscheidend für den Erfolg eines Spiels wie die Pre-Production. Und in kaum einer Phase werden so viele Fehler gemacht, sagt Producer Wolfgang Walk.

Wolfgang Walk

ist selbstständiger Producer.

Wolfgang Walk begann seine Karriere 1995 bei Blue Byte und wirkte u.a. an den Siedler-Teilen 2 bis 4 und an Incubation mit. 2001 wechselte er zu Massive Development, um als Projektleiter die AquaNox-Serie zu verantworten. Seit 2005 ist er freier Producer, unter anderem für Atari, Bitcomposer und gamigo. Momentan arbeitet er an zwei noch nicht angekündigten Titeln.

Die Zeiten, in denen Spiele ohne offizielle Pre-Production gebaut wurden, dürften beinahe vorbei sein. Bei sehr kleinen Titeln mag das ja noch vorkommen, aber selbst bei kleineren Publishern ist die Nachricht inzwischen angekommen, dass eine Pre-Production dabei helfen kann, Qualität, Kosten und Entwicklungsdauer unter Kontrolle zu halten. Sie erleichtert im besten Fall die Kommunikation zwischen Entwickler und Publisher und ist die beste Versicherung dafür, dass sich ein Investment lohnt.

Das bedeutet natürlich nicht, dass eine Pre-Production ganz automatisch zum Erfolg führt. Meine Erfahrung als Producer zeigt mir, dass weitgehend sehr unklare Vorstellungen davon herrschen, wie man die oben angeführten Ziele erreicht. Und im Umkehrschluss bedeutet das: Welche Fehler man dabei machen kann. Die zehn größten Gefahren möchte ich im Artikel nun kurz vorstellen.

1.Producer brauchen wir erst später

Producer sind teuer. Wahrscheinlich sind sie mit die teuersten Mitarbeiter am Spiel. Noch teurer wird es eigentlich nur, wenn auf sie verzichtet wird. Ich werde immer mal wieder für ein Spiel geheuert, das die Pre-Production bereits hinter sich hat. Diese Spiele haben alle eins gemeinsam: Sie stecken bereits immer in massiven Schwierigkeiten.

Die Gesamtarbeit, die ein Producer in ein Spiel investiert, fällt im besten Fall zu mehr als der Hälfte in die relativ kurze Pre-Production. Dies ist die Phase, in der die wichtigen Entscheidungen getroffen werden. Dies ist die Zeit, in der das Team sich wirklich beweisen muss. Und dies ist die Zeit, in der die Arbeit des Teams am engsten überwacht werden muss. Nicht, weil die Teams dazu neigen, auf der faulen Haut zu liegen. Sondern weil auch kleinste Abweichungen von den Vorstellungen des Publishers später zu großen Schwierigkeiten führen können. Weil ein über-

Positiv-Beispiel: Bei **Anno 1404** war von Anfang an ein Producer dabei, in die Pre-Production wurden mehr als sechs Monate investiert. Das Aufbauspiel erschien pünktlich, blieb im Budget-Rahmen und erreichte darüber hinaus auch die angestrebte Qualität.

sehenes Risiko sehr teuer werden kann. Und weil ein Team zu der Zeit vollkommen mit dem neuen Spiel beschäftigt ist und ihm dabei der wichtige Blick von außen häufig abhanden kommt.

Viele Publisher unterschätzen das. Es reicht nicht, einen Produktmanager einen halben Tag pro Woche aufs Team schauen zu lassen. Ein Producer, der sich weniger als drei Tage in der Woche um eine Pre-Production kümmert, kann seinen Job eigentlich nicht zuverlässig erledigen.

2. Den Vertrag gibt's erst nach der Pre-Production

Eine Warnung an Entwickler: Es soll Publisher geben, die Teams erst nach der abgeschlossenen Pre-Production zu zahlen bereit sind. Eine Warnung an Publisher: Jeder Euro, der in der Pre-Production eingespart wird, kostet bis zur Veröffentlichung im Schnitt drei Euro durch Verzögerungen und Qualitätsprobleme.

Am Ende kriegen die Erbsenzähler das, was sie verdienen: ein unfertiges oder zumindest schlechtes Produkt. Es erscheint zu spät und war wahrscheinlich auch zu teuer. Ein weiteres Resultat ist ein demotivierter Entwickler, mit dem man keine dauerhafte, erfolgreiche Geschäftsbeziehung mehr eingehen kann, weil er entweder unmittelbar nach Lieferung insolvent ist oder einen Publisher findet, der ihn künftig besser behandelt. Ausnahmen? Mir sind keine bekannt.

3. Für eine Pre-production reichen 14 Tage

Eine Pre-Production, die den Namen verdient, benötigt etwa ein Drittel der anfangs veranschlagten Ressourcen des Gesamtprojekts. Je nach Projektdauer kann da schon ein erklecklicher Zeitraum zusammenkommen. Bei einer Produktionszeit von zwei Jahren dürfen also auch schon mal acht Monate in die Pre-Production gehen. Und natürlich müssen die Entwickler in dieser Zeit bezahlt werden. Ein seriöser Publisher investiert also circa 25 bis 30 Prozent des Entwicklungsbudgets in die Pre-Production. Was kriegt der Publisher dafür? Zuweilen nicht viel Vorzeigbares: einen Prototypen, in dem vielleicht nur ein paar Klötzchengrafiken hin und hergeschoben werden. Vielleicht gar nur ein paar Excel-Sheets, ein paar Grafikentwürfe, oder ein Brettspiel mit Legofiguren. All das kann am Ende einer Prototypphase stehen.

Was er aber noch bekommt: eine genaue Planung, ein detailliertes Design, eine Ressourcenabschätzung, die keine Varianz von +100/-50% mehr hat, sondern jetzt bei +25/-20% liegt. Man frage mal die Buchhaltung, was das bedeutet. Außerdem haben Publisher und Entwickler jetzt eine gemeinsame Vision. Was mich zum nächsten Punkt bringt.

4. Visionen sollte man behandeln lassen

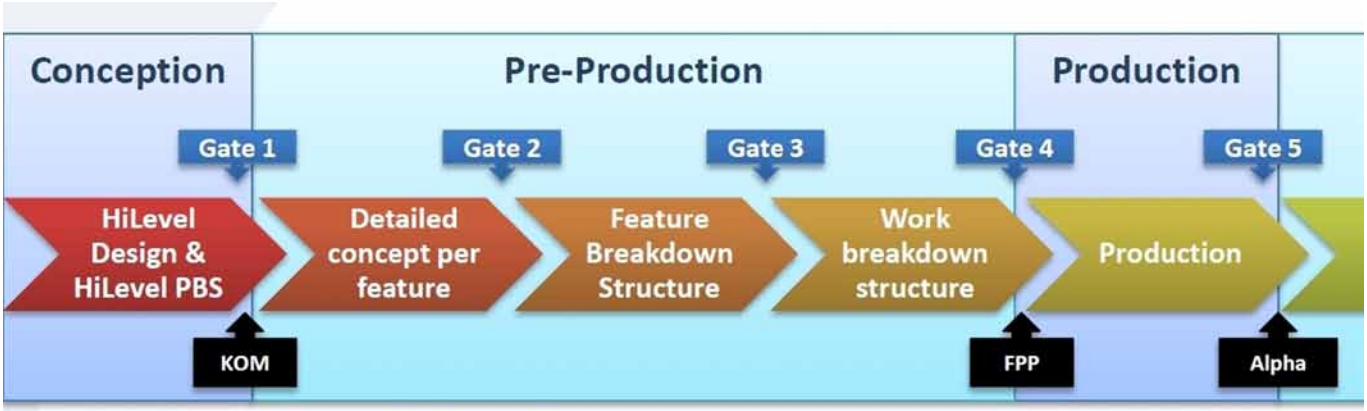
Stimmt! Aber auch dazu ist eine Pre-Production gut. Ganz am Anfang hat der Entwickler eine Vision des Spiels. Oder der Publisher, der sich dann einen Entwickler sucht (und hoffentlich auch einen Produzenten). Und mit viel Glück stimmen diese Visionen sogar überein. Aber eines gilt fast immer: Die Vision VOR der Pre-Production wird behandelt, umgebaut, schärfer formuliert – und am Ende der PP wird sie sich geändert haben. Vor allem wird sie eines geworden sein: realistischer. Und sie wird dann auch ganz sicher zwischen Publisher und Entwickler geteilt.

Diese gemeinsame Vision des Spiels ist eines der wichtigsten Ziele einer Pre-Production! Es gibt mehrere dieser Ziele – und eines der ersten ist es, die Ziele einer Pre-Production festzulegen. Klingt seltsam, ist aber so.

5. Man kann es mit Zieldefinitionen auch übertreiben!

Beginnen wir mal am Ende: Irgendwann muss ja mal überprüft werden, ob die Pre-Production ein Erfolg war oder nicht. Wenn man sich hier nicht auf eine Pi-mal-Daumen-Entscheidung verlassen möchte (was leider allzu häufig geschieht), benötigt man Kriterien. Diese Kriterien müssen irgendwann einmal definiert werden. Und wann erledigt man diese lästige Arbeit? Natürlich in der Pre-Production! Deshalb unterteile ich eine solche auch gern in zwei Teile: eine Konzeptphase und eine Prototypenphase. Am Ende der Konzeptphase muss definiert sein, welche Kriterien am Ende der Pre-Production erfüllt sein müssen. Hier fehlt der Platz, um auch nur annähernd detailliert genug aufzuzählen, was zu diesen Kriterien alles gehört: ein detailliertes Pflichtenheft für die Prototypen, Definitionen der verlangten Design- und Planungs-Granularität, natürlich die Vision des Spiels, eine detaillierte Risiko-Analyse, Teamanalyse, ein Pflichtenheft für den Publisher etc. Und bevor diese Kriterien nicht erfüllt sind – ALLE! – ist die Pre-Production auch nicht abgeschlossen. Und schon gar nicht erfolgreich. Vorher vermittelt sie höchstens einen ziemlich »uneducated guess«. Mit Betonung auf die erste Silbe!

Negativ-Beispiel: Das Lizenzspiel **Schlag den Raab** war zwar kommerziell erfolgreich, hatte jedoch eine viel zu kurze Pre-Production-Phase ohne Producer. Das Team hatte dementsprechend während der Vollproduktion dann auch mit vielen Problemen zu kämpfen.



Am Schema des **Feature Driven Stage-Gate Prozesses** lässt sich die Bedeutung der Pre-Production-Phase sehr gut erkennen.

6. Detailliertes Design ist was für Feiglinge

»So früh kann man noch nicht alle Features detailliert durchdesignen!« Hätte ich jedes Mal, wenn ich diesen Satz von Entwicklern gehört habe, einen Euro bekommen ...

Tatsache ist: Man kann. Zumindest detailliert genug, um ein paar Dinge sicher entscheiden zu können: Existieren technologische Risiken in diesem Feature? Sind seine Auswirkungen auf den Rest des Spiels beherrschbar? Wie aufwändig wird die Implementierung? Haben wir die Leute, es auch zu implementieren? Ist sein Balancing beherrschbar? Ist das Feature den Aufwand denn überhaupt wert?

Wenn Ihr Entwickler behauptet, dass er das in der Pre-Production noch nicht abschätzen kann, dann gibt es zwei Möglichkeiten: Er kann's entweder nicht (sehr schlecht), oder er hat keine Lust auf diese zugegebenermaßen mühselige Arbeit (verständlich). Machen Sie ihm klar, dass er die obigen Fragen zu beantworten HAT, damit sie die Pre-Production als beendet ansehen. Dann finden Sie heraus, ob der Entwickler dazu in der Lage ist.

7. Planung passt nicht zu Agile

Ein Lieblingsargument jedes Entwicklers ist, das in agilen Methoden keine genaue Planung notwendig sei. Meiner Erfahrung nach ist das Unfug. Keine detaillierte Planung bedeutet nur, dass keine Vorstellung davon herrscht:

- welche Aufgabe ein Feature überhaupt hat
- wie groß es ist
- wer es in welchem Zeitrahmen entwickeln soll
- welche Auswirkungen auf Gameplay und Balancing es hat
- welche technologischen Risiken darin stecken
- welche zeitlichen Risiken darin stecken

Und das bedeutet nichts anderes als: Dieses Feature ist ein großes Entwicklungsrisiko und muss mit hoher Priorität angegangen werden! Einfache Regel: Ein Task, der nach Ende der Pre-Prodcution noch mit mehr als zwei Manntagen im Projektplan steht, bleibt ein Risiko und muss als solches geflaggt und möglichst bald adressiert werden. Ein paar davon sind hinnehmbar, aber sie müssen die Ausnahme sein.

Agile kann nicht bedeuten, dass man auf Sicht durchs Projekt steuert, was in etwa dem Versuch vergleichbar ist, auf einem Supertanker im Nebel und mit abgeschaltetem Sonar durch die Arktis zu navigieren. Agile ist eine Entwicklungsmethode. Sie bedeutet in diesem Zusammenhang stark verkürzt lediglich, dass die Planung keine Vorgabe über die Reihenfolge der Entwicklung ist. Agile fordert auf Seiten des Publishers Flexibilität bei der Abwicklung der Milestones/Releases und einen erhöhten Kommunikationsbedarf zum Entwickler hin. Aber eine Pre-Production ohne die vollständige und detaillierte quantitative Erfassung der zu leistenden Arbeit kann logischerweise keine auch nur halbwegs seriöse Aussage über den Release-Termin treffen.

8. Ein guter Prototyp ist schon die halbe Miete

Einige Prototypen sehen schon richtig gut aus – gerade dann, wenn die Pre-Production über längere Zeit läuft und schon einiges an Grafik-Assets vorhanden ist. Schließlich zählt die Installation der Grafikpipeline meist zu den kleineren Herausforderungen einer Spieleproduktion.

Publisher neigen zu der Annahme, dass das Spiel jetzt eigentlich schon halb fertig ist. Der Entwickler darf sich hier nicht auf Deals einlassen: Der meiste Code eines Prototypen gehört zur Kategorie »Schnell und schmutzig«. Kaum eine Zeile ist belastbar, wenn noch weitere Features hinzukommen, die Mechanik komplizierter wird, die Speicherbelastung steigt etc. Es ist eher die Regel als die Ausnahme, dass es schneller und billiger wird, den Code nach der Pre-Production noch mal neu zu schreiben. Man weiß jetzt, wie es geht, man kennt die Fragestellungen, und am Ende wird das Spiel wesentlich weniger Zeit beim Debuggen verbringen. Tipp für Entwickler: Reden Sie in Gegenwart des Publishers möglichst häufig darüber!

9. Die Pre-Production identifiziert alle Risiken

Ein weitverbreiteter Irrtum bei Publishern ist es, dass eine Pre-Production alle wichtigen Risiken adressiert und damit auslöscht. Dies ist ein Irrtum, weil erstens das Controlling der Pre-Production seitens des Publishers meist viel zu

oberflächlich ist und zweitens auch die sorgfältigste Pre-Production nicht alle Risiken identifizieren oder sofort auflösen kann. Allerdings heißt dies natürlich nicht, dass man es dann ja auch auf sich zukommen lassen kann. Wer sich jemals bei der Wahl der Middleware vergriffen hat, oder glaubte, er könne die zwei AI-Coder, die er in sechs Monaten braucht, dann auch sicher genau im richtigen Moment finden, der weiß, wovon ich spreche.

Eine exakte Risiko-Definition, die jeden auch nur halbwegs wahrscheinlichen Fall behandelt, der während der Entwicklung eintreten kann, gehört unabdingbar zu einer Pre-Production dazu (und endet mit ihr natürlich nicht). Als Producer befrage ich dazu ganz gern auch mal den Coder oder Grafiker am Ende der Nahrungskette. Diese Leute, die es ja ausbaden müssen, haben oft ein sehr feines Gespür für Dinge, die gerade von Vorgesetzten unbemerkt schief gehen.

10. No-go-Entscheidung führt zum Aus für Entwickler

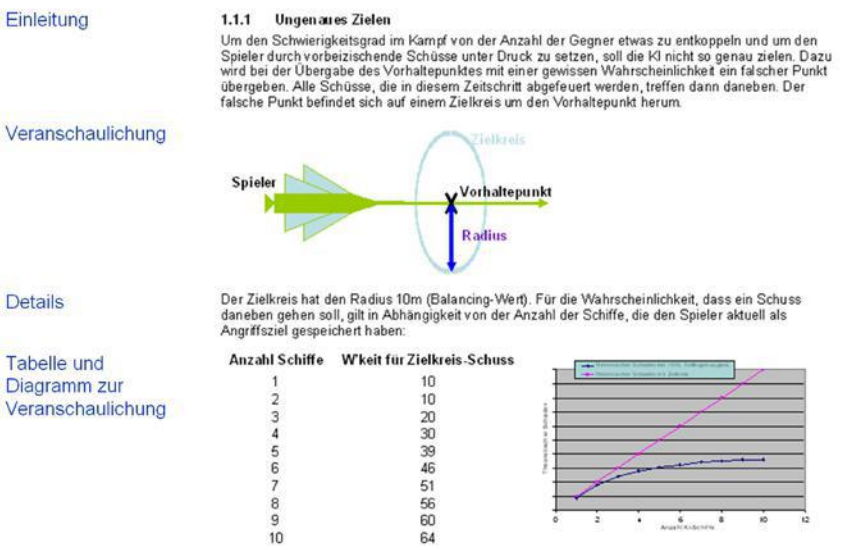
Am Ende jeder Pre-Production folgt eine Go/No-go-Entscheidung des Publishers. Ab diesem Zeitpunkt beginnt für die meisten Entwickler das große Zittern, denn eine No-go-Entscheidung wirft sehr viele von ihnen in eine existenzielle Krise finanzieller Art. Dies gilt offensichtlich immer dann, wenn die Pre-Production unsauber oder unvollständig abgeschlossen wird. Denn einer unsauberen Pre-Produciton folgt niemals ein gutes Produkt!

Aber selbst dann, wenn die Pre-Production erfolgreich war, kann es eine No-go-Entscheidung geben, denn der Erfolg einer Pre-Production wird ausschließlich dadurch definiert, dass der gesamte Kriterienkatalog korrekt beantwortet werden konnte.

Wenn am Ende für den Publisher die Erkenntnis steht, das Aufwand und Ertrag für dieses Spiel nicht in Einklang zu bringen sind, dann folgt zwar das Stoppschild. Aber zwei Drittel eines Budgets rechtzeitig einzusparen, bevor man es vollständig zum Fenster rauswirft, darf durchaus als Erfolg gewertet werden.

Der Entwickler hat also als Dienstleister eine Erkenntnis geliefert, die den Publisher vor einer sechs- bis achtstelligen Fehlinvestition schützt. Eigentlich hätte er also einen sehr guten Grund, weiter mit dem Entwickler zusammenzuarbeiten! Leider ist diese Erkenntnis bei vielen Publishern noch nicht gewachsen. Sie alle wären sehr gut beraten, dem Entwickler während der Pre-Production schon Folgendes klar zu machen: Wenn wir auf Basis einer sauber durchgeführten Pre-Production eine No-go-Entscheidung treffen, dann sehen wir das als Erfolg und für euch geht's weiter. Nichts ist gefährlicher für einen Publisher als ein Entwickler, der anfängt Daten zu schönen, Risiken zu verschweigen und potemkinsche Dörfer hochzuziehen, weil er Angst um seine Existenz hat.

Eine saubere Pre-Production liefert eben nicht nur eine Grundlage für die weitere Entwicklung.


Einleitung
1.1.1 Ungenaues Zielen
Um den Schwierigkeitsgrad im Kampf von der Anzahl der Gegner etwas zu entkoppeln und um den Spieler durch vorbeizischende Schüsse unter Druck zu setzen, soll die KI nicht so genau zielen. Dazu wird bei der Übergabe des Vorhaltepunktes mit einer gewissen Wahrscheinlichkeit ein falscher Punkt übergeben. Alle Schüsse, die in diesem Zeitschritt abgefeuert werden, treffen dann daneben. Der falsche Punkt befindet sich auf einem Zielkreis um den Vorhaltepunkt herum.
Veranschaulichung
Zielkreis
Spieler
Vorhaltepunkt
Radius
Details
Der Zielkreis hat den Radius 10m (Balancing-Wert). Für die Wahrscheinlichkeit, dass ein Schuss daneben gehen soll, gilt in Abhängigkeit von der Anzahl der Schiffe, die den Spieler aktuell als Angriffsziel gespeichert haben:
Tabelle und Diagramm zur Veranschaulichung
Anzahl Schiffe W'keit für Zielkreis-Schuss
1 10
2 10
3 20
4 30
5 39
6 46
7 51
8 56
9 60
10 64

Auch in der Pre-Production-Phase lassen sich selbst Gameplay-Details schon komplett durchdesignen – hier ein Beispiel aus dem **Game Design Dokument** von Darkstar One.

Paraworld sah bereits als Prototyp hervorragend aus. Das Spiel erschien trotzdem mit fast zweijähriger Verspätung, kostete viel zu viel und bedeutete letztendlich das Aus für Entwickler SEK und Publisher Sunflowers.

Sie ist zunächst einmal eine Entscheidungsgrundlage dafür, ob sich die Entwicklung überhaupt lohnt! Es ist nicht nur fair, sondern sogar vernünftig, den kompetenten Lieferanten dieser Entscheidungsgrundlage nicht von der späteren Entscheidung abhängig zu machen.

Wolfgang Walk

1.000 Schwerter in 2 Tagen
Generierung von Waffen-Modellen

Nicht nur Bilder, sondern auch Ingame-Assets lassen sich automatisiert generieren. In diesem Workshop erklärt Jochen Winkler Schritt für Schritt, wie Sie nahezu unbegrenzt viele Waffen erstellen – und dabei Fallstricke vermeiden.

Jochen Winkler

ist selbstständiger Game Artist.

Nach sechs Semestern Mathematikstudium wechselte Jochen Winkler auf die Games Academy in Berlin und absolvierte die Ausbildung zum Game Artist. Seitdem ist er als Grafiker tätig und nutzt seine Doppelqualifikation, um ungewöhnliche Grafikprobleme mit Algorithmen und mathematischen Formeln zu lösen. Seit Anfang 2011 ist er selbstständig und kümmert sich nach eigener Beschreibung um »Spiele und algorithmische Grafik«. Jochen ist unter info@jochen-winkler.com erreichbar und freut sich über Anfragen – egal ob Sie 3.000 Aliens oder einfach nur Hilfe beim Durcharbeiten dieses Tutorials benötigen.

WWW
makinggames.de
- Cinematics aus Ingame-Assets

Spätestens seit dem Erfolg von Diablo 1 wissen wir: Waffensammeln macht Spaß! Irgendwann ist diese Erkenntnis allerdings mit der Branche durchgegangen. Angehende Game Designer wünschen sich meist mehr Waffen, als ein Team jemals umsetzen könnte. Marketingpartner wünschen sich große Zahlen, um sie auf die Packungsrückseite zu drucken. Und die Geldgeber? Die freuen sich ebenso, denn »Spielspaß« kann man nicht per Vertrag einfordern. »800 Waffen« allerdings schon.

Doch der Erfolg von Micropayment hat der Situation einen echten Sinn verliehen. Seit wir virtuelle Waffen für bares Geld verkaufen, gilt tatsächlich: Viele Waffen bringen auch viel Geld!

In diesem Workshop werden Sie eine möglichst große Anzahl von Schwertern in möglichst kurzer Zeit generieren. Die technische Seite ist dabei relativ einfach, schwieriger ist es, die generierten Schwerter nicht generisch wirken zu lassen. Die folgenden Beispiele wurden allesamt mit Maya erstellt. Sie können jedoch mit jedem 3D-Paket arbeiten, das die Möglichkeit zum Skripten bietet.

Je nachdem, ob Sie für ein 2D-Spiel oder für einen komplexen 3D-Titel arbeiten, werden Ihnen bestimmte Teile des Tutorials wichtiger sein.

Die Grundlagen

Wie schon in meinem letzten Workshop über automatische Content-Generierung in Ausgabe 01/2011 entsteht ein Teil der Vielfalt mittels Durchtauschen. Doch Vorsicht: Natürlich können Sie aus 10 Klingen, 10 Griffen und 10 Farben exakt 1.000 Kombinationen erstellen. Besonders abwechslungsreich wären diese allerdings nicht. Deshalb investieren wir etwas mehr Handarbeit.

Stunde 1: Bildrecherche

Bei der Bildrecherche im Internet lohnt es sich, auch die zugehörigen Texte zu beachten. Sobald Ihnen ein interessanter Begriff auffällt, können Sie ihn für eine weitergehende Suche nutzen. Fachbegriffe wie »Fehlschärfe« oder »Kopesch« liefern Ihnen auf Dauer vielfältigere Ergebnisse als die zwanzigste Seite zum Begriff »Schwert«.

Stunde 2 und 3: Modellieren der ersten Klingen

Ob Sie tatsächlich zwei Stunden für das Modellieren von mehreren Klingen brauchen, hängt von Ihrem Projekt ab. Low-Poly-Modelle gehen natürlich schneller von der Hand, Modelle für Großaufnahmen (wie die Bilder für diesen Artikel) dauern etwas länger. Folgende Punkte gilt es zu beachten:

- Wählen Sie extreme Modelle. Natürlich erkennt man eine auffällige Klinge in Kombination mit verschiedenen Griffen und Texturen schnell wieder. Doch für 1.000 unterschiedliche Schwerter brauchen wir auch extreme Unterschiede. Um das Problem mit dem Wiedererkennungseffekt kümmern wir uns später.
- Der Ansatz, mit dem die Klinge im Griff steckt, sollte eine gewisse Breite nicht überschreiten. Nicht jeder Ihrer Griffe wird breit sein, und auch breite Klingen sollten schließlich in jeden Griff hineinpassen (**Abbildung 1**).
- Achten Sie beim Modellieren auf das »Gewicht« der Klinge (**Abbildung 2**). Eine kurze Dolchklinge auf dem Griff eines Zweihänders würde im finalen Zustand absurd aussehen.
  Ihre Schwerter müssen nicht gleich lang sein, aber sie sollten in etwa gleich schwer wirken.
- Modellieren Sie keine Gravuren, aber definieren Sie Bereiche, in denen solche passend wären.
- Stellen, die aus unterschiedlichen Materialien bestehen sollen, teilen Sie am besten in mehrere Objekte auf. Ordnen Sie diese stets als Klinge mit Unterobjekten an (**Abbildung 3**). Später beim Skripten können Sie ohne Aufwand die gesamte Objektgruppe auf einmal auswählen.
- Erstellen Sie zunächst noch keine UV-Maps.

Die Klinge definiert die Form eines Schwertes am stärksten, gleichzeitig ist sie am einfachsten zu modellieren. Erstellen Sie daher ruhig einige Klingen mit ausgefallenen Varianten mehr.

Stunde 4: Modellieren der Griffe

Generell gilt das Gleiche wie bei den Klingen. Erstellen Sie zunächst einige extrem unterschiedliche Modelle. Halten Sie die eigentliche Griffffläche so einfach wie möglich. Mit Hilfe von Displacement-Maps können Sie die Geometrie dieser Fläche unabhängig vom restlichen Griff durchtauschen.

Für die gepolsterten Teile der Griffe sollten Sie schon UV-Maps erstellen. Eventuell müssen Sie die Griffffläche aufgrund des Offsets der Displacement-Maps dünner oder dicker anlegen als gedacht. Um dies zu testen, können Sie zwischendurch zum nächsten Schritt springen und eine einzelne Displacement-Map erstellen. Sofern Sie nur Normal-Maps verwenden, haben Sie dieses Problem natürlich nicht.

Stunde 5: Griff-Texturen und erste Metallfarben

Für den Anfang genügt es, ein Schwert in zwei Oberflächenarten zu unterteilen: die Metallflächen und den gepolsterten Teil des Griffs. Natürlich haben echte Schwerter oft mehr als zwei Oberflächen. Es gibt beispielsweise stählerne Klingen auf einer vergoldeten Parierstange. Doch wenn Sie das Zusammenstecken von Griff und Klinge etwas kaschieren wollen, sollten Sie für beide dasselbe Metall verwenden.

Von Schwert zu Schwert darf (und sollte) sich das Material natürlich verändern. Erstellen Sie also zwei Materialgruppen: Metalle (Stahl, Eisen, Kupfer usw.) und Griffpolsterungen (etwa Leder und Stoffpolsterungen).

Verzichten Sie bei den Metallmaterialien noch auf alles, was von der UV-Map abhängt. Verteilen Sie also Farben und Reflexionswerte, verzichten Sie aber noch auf Texturen. Bei den Griffen hingegen ist bereits alles möglich.

Je nach Natur Ihres Projekts können andere Gruppen nötig werden. Es wäre etwa eine für Holzmaterialien nötig, um eine große Auswahl an Äxten oder Speeren zu generieren. Andere Projekte benötigen eventuell glühende Materialien für Magie- oder Laserschwert-Effekte. Versuchen Sie in diesen Fällen, die Anzahl der Materialien auf jeder Waffe gering zu halten. Bemühen Sie sich außerdem, besonders dominante Materialien auf allen Teilen der Waffe zu wiederholen.

Displacement-Map oder doch nur Normal-Map?

Auf Normal oder Displacement-Maps sollten Sie auf keinen Fall verzichten. Besonders an der Klinge haben Sie sonst kaum Möglichkeiten, die Geometrie durch abwechslungsreiche Texturen zu verändern. Vielfarbige Metalle gibt es schließlich selten, nur Specular und Reflection-Map könnten noch komplexere Muster abbilden.

In vielen Fällen haben Sie keine Wahl: Manche Engines können ausschließlich Normal-Maps verarbeiten. Sofern Sie die Möglichkeit haben, sollten Sie statt der Normal-Maps lieber Displacement-Maps wählen. Das liegt weniger an den

Abbildung 1: **Breite Klingen** werden im Ansatz gestaucht, damit Sie auch auf schmale Griffe passen.

Abbildung 2: Die rechte Klinge gehört zu den längsten, bleibt aber dünn, damit sie **nicht zu »schwer«** wirkt.

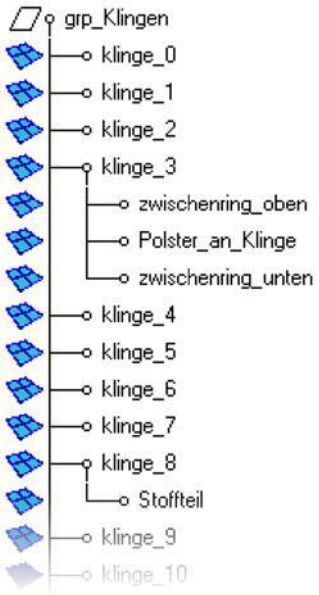


Abbildung 3: Klingen mit mehreren Materialien sollten stets **hierarchisch** angelegt werden.

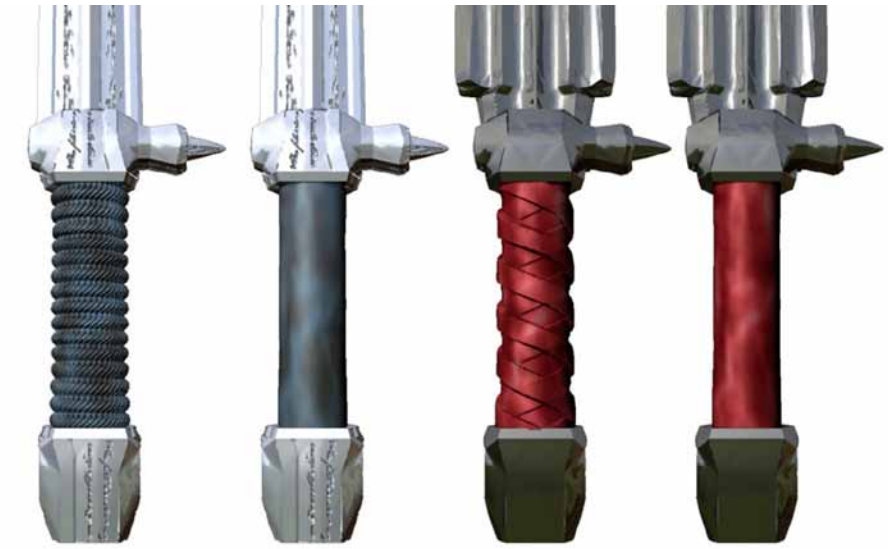


Abbildung 4: Griffstücke mit und ohne **Displacement Map** für die Veränderung der Geometrie.

Klingen als am Griff: Displacement-Maps verändern auch die Silhouette der Objekte (**Abbildung 4**). So scheinen dann sogar identische Griffstücke noch Unterschiede in der Form zu haben.

Eventuell hat der Spieler aber nie die Möglichkeit, Griffstücke der Schwerter nah genug zu bewundern – insbesondere wenn keine Nahansicht für das Inventarsystem integriert ist. In diesem Fall können Sie sich natürlich den ganzen Aufwand sparen.

Egal ob Sie mit Normal oder Displacement-Maps arbeiten: Testen Sie diese möglichst früh in Ihrer Engine! In den meisten Fällen weicht die Darstellung stark von der ab, die Sie in Ihrer 3D-Software sehen.

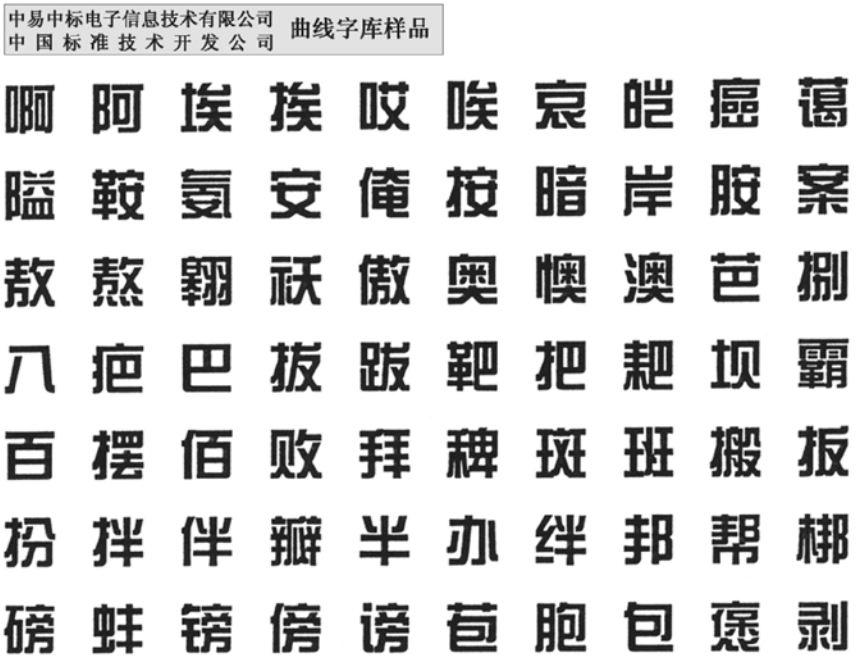


Abbildung 5: Mit der **Schriftsammlung Ihres Vertrauens** sind ein paar Gravuren schnell beisammen. Asiatische Schriften sollten vorher vom Übersetzer geprüft werden: Nicht nur eine ungeahnte Bedeutung kann Probleme bereiten – vermutlich wirkt der taiwanesische Zeichensatz für Festlandchinesen so wenig mittelalterlich wie beispielsweise Arial oder Comic Sans für uns Europäer.

Stunde 6 bis 8: Gravuren und die UV-Maps der Klingen

Klingen haben Gravuren. Das können zum Beispiel elfische Inschriften oder keltische Knoten sein, dämonische Fratzen, geritzte Flüche, Engelsfiguren oder Totenköpfe.

Sie wollen solche Texturen für jede Klinge nutzbar machen? Wenn Sie die Texturen auf die Form der einzelnen Klingen anpassen, müssten Sie bei nur 5 Gravur-Texturen und 12 Klingen ganze 60 Texturen erstellen. Wesentlich effizienter: Sie erschaffen für jede der 12 Klingen je ein UV-Set, das an die Platzaufteilung der Gravuren angepasst ist. Vergessen Sie dabei nicht, dass UVs auch über die Texturen hinausragen dürfen. Um allerdings Kratzer, Rost oder getrocknete Blutflecken darzustellen, sind diese UV-Sets ungeeignet.

Falls Sie in Ihrem jeweiligen 3D-Programm ausgesprochen fit sind, empfehle ich multiple UV-Maps. Eine für die Gravuren, eine zweite für oben genannte Verschmutzungen. Je nach Geschmack können auch weitere UV-Maps erstellt werden, um etwa gefalteten Stahl darzustellen.

Wenn Sie unter Zeitdruck stehen oder falls Sie noch keine tiefergehende Erfahrung mit Ihrem 3D-Programm haben, können Sie diesen Schritt einfach überspringen. Das gilt besonders, wenn Sie für einen 3D-Titel arbeiten. Die auswechselbaren Texturen auf verschiedenen UV-Maps verschönern die Schwerter deutlich, müssen jedoch aufwendig auf eine einzige UV-Map gebakt werden, wenn es an den Export geht. Fragen Sie allerdings vorher Ihren Programmierer. Mit sehr viel Glück stört ihn eine zweite UV-Map überhaupt nicht und Sie können auf das Baken verzichten.

Bei den Schwertern in den Beispielbildern wurde übrigens auf multiple Texturen verzichtet. Aufgrund der Projektanforderungen erhielten die Griffe nicht einmal mehr eine sinnvolle UV-Map.

Stunde 9 bis 12: Varianten von Klingen und Griffen

Wenn die Texturen getestet und alle UV-Maps angelegt sind, erstellen Sie Varianten der Griffe und Klingen. Konzentrieren Sie sich zunächst auf die Klingen, die Silhouette der Klinge dominiert schließlich das gesamte Schwert.

Wenn Sie die Möglichkeiten Ihres 3D-Programms voll ausschöpfen, können Sie manche Variationen in weniger als einer Minute erstellen. Verlängern und verkürzen Sie Fehlschärfen. Geben Sie den Klingen eine konkave, eine konvexe und eine gerade Silhouette. Verbreitern Sie Klingen an Spitze oder Basis. Wichtig ist, dass sich die Silhouette verändert.

Die UV-Maps für Gravuren können Sie unverändert lassen, die UV-Maps für Verschmutzungen können Sie entweder neu erstellen oder die geringen Verzerrungen in Kauf nehmen.

In diesem Schritt löst sich auch das Problem mit den »zu einzigartigen« Klingen. Wenn Sie nur eine einzige Kopesch-Klinge haben, fällt diese dem Spieler als einzigartiges, schlecht integriertes Element auf. Wenn Sie vier unterschiedliche Varianten von Kopesch-Klingen modellieren, erkennt der Spieler ein Schwert zwar als Kopesch, hält die Varianten des Schwerts jedoch für einen sinnvollen Aspekt der Spielwelt. Dasselbe gilt im Übrigen für Griffe: Ein einzelner Griff mit auffallendem Drachenkopf ist ein Problem. Drei Griffe mit auffallenden Drachenköpfen repräsentieren eine Kultur.

Wenn Sie die Technik mit diesem Workshop nur ausprobieren, sollten Sie am Ende zumindest 13 Klingen haben. Für echte Projekte empfehle ich aber 23 oder mehr Varianten. Griffe benötigen Sie nicht ganz so viele, 11 Variationen sollten zum Testen ausreichen.

Puristen möchten vielleicht auch das Erstellen der Klingenvarianten automatisieren. Kunstgriffe wie Blendshapes, automatisierte Latices oder gar skriptgenerierte Stacheln und Klingenwellen sind natürlich möglich. Aber ganz ehrlich: In dieser Situation liefert Handarbeit immer noch das beste Ergebnis.

Stunde 13 bis 15: Skripten für den Export

Im Workshop zu Beginn von 2011 ging es ausschließlich darum, Bilder von den generierten Kombinationen zu rendern. Wenn Sie lediglich Bilder benötigen, können Sie tatsächlich auf dieselbe Methode wie in Ausgabe 01/2011 beschrieben zurückgreifen. In vielen Fällen werden jedoch exportierte 3D-Modelle benötigt. Und spätestens hier kommen Sie um das Skripten nicht mehr herum. Die nötigen Befehle lassen sich jedoch schnell recherchieren. Sie müssen wissen, wie eine for-Schleife funktioniert, wie Sie Objekte selektieren, wie sie die Materialien durchtauschen können und wie Sie über einen Skriptbefehl Ihren Exporter starten.

Dieser letzte Schritt ist wohl der aufwendigste in der Prozedur: Denn bei normalen Exporten müssen je nach Projekt, Exporter und Engine unterschiedliche Optionen gewählt werden. Recherchieren Sie daher die nötigen Befehle und stellen Sie mit ein paar Tests sicher, dass die per Skript exportierten Objekte genauso fehlerfrei im Spiel ankommen wie jene, die per Hand exportiert wurden.

Systematisch jede mögliche Kombination speichern?

Am naheliegendsten sind mehrere ineinanderverschachtelte for-Schleifen. Jede davon zählt ein Element des Schwertes hoch – so wird sorgfältig jede der möglichen Kombinationen zusammengestellt. Die Reihenfolge hat System:

Klinge 1 mit Griff 1 mit Metall 1 mit Grifffarbe 1,
Klinge 1 mit Griff 1 mit Metall 1 mit Grifffarbe 2,
Klinge 1 mit Griff 1 mit Metall 1 mit Grifffarbe 3,
Klinge 1 mit Griff 1 mit Metall 2 mit Grifffarbe 1,
... und so weiter.

Der Nachteil: Wenn Sie nur die ersten 300 Schwerter exportieren, haben Sie weniger Variation als möglich wäre. Dasselbe gilt für die angepeilten 1.000 Schwerter. Denn bei der riesigen Anzahl an Kombinationen sind es eben auch »nur die ersten« 1.000 Stück. Damit die Auswahl unserer Schwerter weniger homogen ausfällt, wählen wir eine andere Methode.

Maximale Abwechslung

Verwenden Sie die Modulo-Funktion, um in einer einzigen for-Schleife in jedem Schritt jedes einzelne Element zu wechseln. Die Grundkonstruktion sieht (als Pseudo-Code) wie folgt aus:

```
for(int i = 0; i < mengeDerZuErstellendenSchwerter; i++){
    leere die Auswahl;
    wähle Klinge Nr (i % klingenAnzahl);
    wähle Griff Nr (i % griffAnzahl);
    weise Metall Nr (i % metallAnzahl) zu;
    weise GriffMaterial Nr (i % griffMaterialAnzahl) zu;
    Exportiere die Auswahl;
}
```

Anstelle (oder zusätzlich zu) der Export-Anweisung können sie Bilder auch rendern lassen, ganz nach den Anforderungen Ihres Projekts.

Um die gleiche Konstruktion konkret in Mayas Skriptsprache MEL umsetzen zu können, müssen Sie einige Kleinigkeiten bei den Texturen vorbereiten. Erstellen Sie zunächst je ein Platzhaltermaterial für Metall und für Griffpolster. Weisen Sie die beiden Platzhalter den jeweiligen Polygonflächen zu und benennen Sie die Shading-Groups (nicht die Materialien selbst) »metal_SG« und »griff_SG«. Ihre bestehenden Materialien benennen Sie »metallMaterial_0«, »metallMaterial_1« usw.; die für die Griffpolsterung »griffMaterial_0«. Die Displacement-Materialien werden ab »metallDisplacement_0« bzw. »griffDisplacement_0« nummeriert. Das Skript in **Abbildung 6** geht davon aus, dass zu jedem Griffmaterial genau eine einzige Displacement-Map gehört, dass sich jedoch die Metall-Displacements (also die Gravuren) frei mit den unterschiedlichen Metallen kombinieren lassen.

Sich häufig wiederholende Kombinationen

Möglicherweise fällt Ihnen beim Exportieren oder Rendern auf, dass bestimmte Elemente immer nur miteinander auftauchen. Vielleicht kommt die lange Klinge immer nur in Gold vor

Was ist Modulo?

Erinnern Sie sich noch an das Teilen in der Grundschule? Als das Ergebnis von »11 : 3« noch »3, Rest 2« war? Weil die Schüler noch keine Nachkommastellen kennen, bleibt beim Teilen oft ein Rest übrig. Modulo ist eben dieser Rest. »11 Modulo 3« ergibt 2, »3 Modulo 2« ergibt 1, »4 Modulo 2« ergibt 0. In den meisten Programmiersprachen wird Modulo mit dem »%«-Zeichen abgekürzt. Die Schreibweise ist also beispielsweise »11 % 3«.
Wofür wird Modulo in unserem Fall verwendet? Wenn Sie etwa 13 Klingen modelliert haben und das 14. Schwert zusammensetzen, werden Sie bei den Klingen wieder von vorn beginnen wollen. Das Modulo-Zeichen erspart ihnen dabei einige Hilfskonstruktionen. Wenn Sie dem Schwert 134 beispielsweise die Klinge 134 % 13 geben, entspricht das der Klinge Nummer 4. Doch Vorsicht: Das Schwert 130 hat die Klinge »Nummer 0«, denn 130 ist glatt (und ohne Rest) durch 13 teilbar.

```
int $mengeDerZuErstellendenSchwerter = 1000;
int $klingenAnzahl = 17;
int $griffAnzahl = 11;
int $metallMaterialAnzahl = 9;
int $griffMaterialAnzahl= 5;


int $i;
for($i = 0; $i < $mengeDerZuErstellendenSchwerter; $i++){
    select -clear;
    select -add („klinge_“ + ($i % $klingenAnzahl));
    select -add („griff_“ + ($i % $griffAnzahl));

    connectAttr -force („metallMaterial_“ + ($i % $metallMaterialAnzahl)+
„.outColor“) metall_SG.surfaceShader;
    connectAttr -force („metallDisplacement_“ + ($i %
$metallMaterialAnzahl)+ „.outDisplacement“) metall_SG.displacementShader;
    connectAttr -force („griffMaterial_“ + ($i % $griffMaterialAnzahl)+
„.outColor“) griff_SG.surfaceShader;
    connectAttr -force („griffDisplacement_“ + ($i % $griffMaterialAnzahl)+
„.outDisplacement“) griff_SG.displacementShader;

    file
        -options „groups=0;ptgroups=0;materials=1;smoothing=1;normals=1“
        -type „OBJexport“
        -exportSelected („C:/Users/JW/Documents/maya/projects/Artikel_
Schwerter/data/Schwert_“ + $i)
    ;
}
```

Abbildung 6: Das **Skript** geht davon aus, dass zu jedem Griffmaterial genau eine Displacement Map gehört.

oder sie ist ausschließlich mit einem von drei Griffen kombiniert, nie aber mit einem anderen. Dieses Problem entsteht, wenn die Stückzahlen zweier unterschiedlicher Elementgruppen einen gemeinsamen Teiler haben. Wenn Sie etwa 12 Klingen und 8 Griffe (gemeinsamer Teiler 4) haben, wird die Klinge Nummer 1 nur in Kombination mit den Griffen 1 oder 5 auftreten. Noch schlimmer wird es, wenn zwei Elementgruppen exakt dieselbe Stückzahl haben. Es gibt zwei Wege, dieses Problem zu vermeiden. Erschaffen Sie einige einfache Varianten, sodass Sie auf Zahlen ohne gemeinsamen Teiler kommen. Oder ersetzen Sie den Term »($i % $elementAnzahl)« eines einzelnen Elements durch die Zufallsfunktion »(rand $i)«.

Fortgeschrittenen empfehle ich, eine »vorhersehbare« Zufallsfunktion zu nutzen. Diese wird allerdings in Maya schnell unübersichtlich. Der Modulo eines MD5-Wertes kann grundsätzlich genauso eingesetzt werden, ist aber nicht in jeder Skriptsprache verfügbar.

Dateinamen und Zusatzinformationen

Oftmals reicht es nicht aus, einfach nur Hunderte von Objekten zu speichern. So könnten die Spielwerte oder der Name des Schwertes mit der Grafik zusammenhängen. Ein Beispiel: Es gibt in Ihrem Spiel Kälteschaden und Sie haben dafür extra eine Eistextur unter die Metalle gemischt. Demzufolge sollte jedes Schwert mit dieser Textur den Namenszusatz »der Kälte« tragen. Es könnte auch sein, dass jedes Schwert mit Klinge 3, 8 oder 15 besonders viel Schaden anrichtet. Eventuell stoßen Sie auch auf das handfeste technische Problem, dass das Objektformat Ihrer Engine nicht die Verknüpfungen zu den richtigen Texturen speichert.

In jedem Fall müssen Sie zusätzliche Informationen speichern. Dabei gibt es wieder grundsätzlich zwei Möglichkeiten:

- Sie speichern alle Informationen (inklusive der verwendeten Einzelteile) in einer zusätzlichen Datei, beispielsweise als einfachen Text oder im XML-Format. Den Dateinamen der Objekte können Sie dabei einfach hochzählen, etwa »Schwert_0325.obj«. Vielleicht freut das Ihren Programmierer, weil er dann alle Schwerter mühelos einlesen kann. Es ist für Programmierer normalerweise einfacher, wenn nicht jedes Schwert eine eigene Informationsdatei hat, sondern wenn die Infos aller Schwerter in einer einzigen Datei zusammengefasst sind.
- Sie speichern die Informationen im Dateinamen. Es ergibt allerdings wenig Sinn, den String »... der Kälte« oder den Schadenswert direkt in den Dateinamen zu schreiben. Besser ist es, durch Zahlenwerte direkt die verwendeten Einzelteile aufzuzählen. Zusatzinformationen wie etwa der Name oder die Schadenswerte werden erst später im Programmcode hinzugefügt. Verwenden Sie bei dieser Methode Buchstabenkürzel. Wenn Sie aus 1.000 Dateien namens »Schwert_18_4_7_11.obj« alle Dateien mit dem Metallmaterial 9 heraussuchen wollen, stehen Sie vor einem Problem. Sofern Sie Ihre Dateien »Schwert_k18_g4_m7_s11.obj« benannt haben, müssen Sie Windows nur nach dem Dateinamen »*_m9_*« suchen lassen, um alle Objekte mit dem Metallmaterial 9 zu finden.

Stunde 16: Schönheitskorrekturen und Nachbessern

Ihnen sind mittlerweile sicherlich Dutzende Kleinigkeiten aufgefallen, die Sie noch ändern wollen. Passen manche Texturen einfach nicht? Lassen sich möglicherweise manche Klingen stilistisch nicht mit anderen Griffen kombinieren? Ist noch mehr Abwechslung bei den Klingen nötig? Oder wünschen Sie sich sogar ein neues Element wie etwa durchtauschbare Stacheln?

Wenn Sie diesen Workshop das allererste Mal durchgearbeitet haben, werden Ihre Ergebnisse noch nicht so gut aussehen wie Ihre »normalen« Werke. Schmeißen Sie nicht das Handtuch! Denn jede Verbesserung, die Sie jetzt einbauen, betrifft nicht nur ein Schwert – sie betrifft Hunderte!

Jochen Winkler

TIME FOR A NEW ERA

FORGE OF EMPIRES

www.forgeofempires.com

InnoGames

Weltweit den Spieler im Blick
Internationales Community Management

Szymon Wilkosz beschreibt im Artikel, welche Team- und Kommunikationsstrukturen bei Mail. Ru Games bestehen, um die komplexen Aufgaben eines internationalen Community Managements bewältigen zu können.

Szymon Wilkosz

ist Head of Community Mangement bei Mail.Ru Games.

Szymon war zweieinhalb Jahre bei der Icans GmbH beschäftigt und konnte dort bereits praktische Erfahrungen als Community Manager sammeln. Etwa zwei weitere Jahre war er dann bei Mai.Ru Games als Product Manager tätig und kümmerte sich unter anderem um die Monetarisierung der Spiele. Der Blick hinter die Kulissen der Produkte veränderte nach eigener Aussage seinen Blick auf das Community Management, das er heute beim russischen Publisher leitet.

Community Management ist die Bezeichnung für alle Methoden und Tätigkeiten rund um Konzeption, Aufbau, Leitung, Betrieb, Betreuung und Optimierung von virtuellen Gemeinschaften sowie deren Entsprechung außerhalb des virtuellen Raumes. Unterschieden wird dabei zwischen operativen, den direkten Kontakt mit den Mitgliedern betreffenden, und strategischen, den übergeordneten Rahmen betreffenden, Aufgaben und Fragestellungen.«

Diese erst im letzten Jahr vom Bundesverband für Community Management e.V. veröffentlichte Definition beschreibt die Aufgabenfelder des Community Managements in Unternehmen. Durch die sehr allgemein gehaltene Formulierung lässt sie sich auf beinahe jeden beliebigen Bereich und jedes Unternehmen übertragen – so auch auf die Spielebranche. Hierbei werden die Entwickler und Publisher jedoch aufgrund der engen Vernetzung von Produkt und Community vor spezielle Herausforderungen gestellt.

Anhand praxisnaher Beispiele innerhalb der Mail.Ru Games GmbH möchte ich die Funktionsweise und den Aufbau des Community Managements in international agierenden Unternehmen aus der Gaming-Branche aufzeigen und die damit verbundenen Probleme – inklusive einiger Lösungsansätze – erläutern.

Im komplexen Rollenspiel **Drachenkrieg** von Mail.Ru Games kümmern sich die Community Manager nicht um die Umsetzung von Ingame-Events, sondern um deren Kommunikation an die Spieler. Sie fungieren somit als Sprachrohr, während das Product Management die spielinterne Organisation übernimmt.

Fachkräftemangel

Community Management, so wie wir es heute kennen, existiert in der deutschen Spielebranche erst seit einigen Jahren. Dennoch ist der Bedarf an geschulten Kräften aus diesem Bereich immens, was die enorme Anzahl an Stellenausschreibungen belegt. Die Anforderungen an potenzielle Bewerber fallen dabei abhängig vom Unternehmen unterschiedlich aus. In **Abbildung 1** sehen Sie beispielhaft ein aktuelles Anforderungsprofil von Mail.Ru Games.

Im Vergleich zu manch anderem Unternehmen, bei dem als Voraussetzung oft ein abgeschlossenes Studium aufgeführt wird, fallen die Anforderungen von Mail.Ru Games in meinen Augen relativ mild aus. Ein Community Manager muss vor allem eines sein: kommunikativ – und ob jemand diese Fähigkeit besitzt, lässt sich anhand eines Abschlusses nur schlecht feststellen.

Während es in Deutschland genügend Fachpersonal für den deutsch- und englischsprachigen Markt gibt, herrscht auf anderen Märkten Knappheit. Dabei sind nicht einmal Exoten wie etwa China oder Indien gemeint. Betroffen sind Märkte wie Spanien, Frankreich oder Polen.

Doch wo liegt der Ursprung dieses Problems? Ist es die noch junge Berufsbezeichnung, die bist jetzt wenig fruchtbaren Boden bietet und weiterhin mit fehlender gesellschaftlicher Anerkennung zu kämpfen hat? Es besteht tatsächlich das Problem, dass nur sehr wenige Menschen den Job »Community Manager« kennen. Kein Wunder, sucht man doch vergebens nach einer Ausbildungsstelle oder anderweitigen Bildungsmöglichkeit. Studiengänge im Bereich Games decken nur Teile des Themas ab, eine Spezialisierung in dem Feld fehlt.

Oder scheitert es an zu niedrigen Gehältern? Das Durchschnittsgehalt eines Community Managers in der Spielebranche liegt – abhängig natürlich vom Unternehmen, den Qualifikationen und der Erfahrung – momentan zwischen 15.000 und 25.000 Euro Brutto im Jahr, wobei in vielen Unternehmen die Stelle des Community Managers mit Praktikanten oder 400-Euro-Kräften besetzt wird.

Missfallen vielleicht die unregelmäßigen Arbeitszeiten? Community Manager arbeiten nur

in den seltensten Fällen 40 Stunden pro Woche. Meist ist ihre Arbeit mit Überstunden und Home Office verbunden – eine Community schläft schließlich nie.

Ich bin der Meinung, dass noch zwei weitere Faktoren zu den oben genannten die Misere des Mangels beeinflussen. Zum einen ist die Branche sehr jung, wodurch eine öffentliche Wahrnehmung dieses Berufs weitgehend fehlt. Zum anderen sind Fachkräfte mit mehrjähriger Erfahrung in der Spielebranche rar und werden stark umworben. Wie man sieht, sind die Probleme, mit denen das Community Management zu kämpfen hat, sehr vielfältig. Es wird noch einige Zeit dauern, bis geeignete Lösungen gefunden und deren Umsetzung auf entsprechende Strukturen vollzogen werden können.

Das internationale Community Management bei Mail.Ru Games

Um erklären zu können, in welche Prozesse ein neuer Community Manager integriert wird und welche Position er einnimmt, muss zuerst ein Teil der Unternehmensstruktur offengelegt und erläutert werden. Erst dann lassen sich der Aufbau und der Einsatz unseres Community Managements nachvollziehen.

Das Community Management wird bei Mail.Ru Games zentral gesteuert. Das heißt, dass alle 14 Community Manager, die insgesamt acht Sprachen betreuen, vom Hamburger Büro aus operieren. Die interne Kommunikation verläuft dabei in deutscher Sprache. Im Gegensatz zu vielen anderen Unternehmen übernimmt das Community Management bei uns keine aktiven und direkt mit dem Produkt verknüpften Aufgaben, wie zum Beispiel die Organisation von In-Game-Events. Denn allein in unserem Spiel »Drachenkrieg« existieren für uns unzählige Möglichkeiten, aktiv Einfluss auf das Spielgeschehen zu nehmen. So können wir relativ flexibel auf die Wünsche unserer Spieler reagieren und mehrstufige, individuelle Events gestalten und beliebig viele Anpassungen und Erweiterungen vornehmen. Weil dies naturgemäß eine sehr zeitaufwändige Prozedur ist, kann sie ein Community Manager, der vorrangig seine Community betreut, nicht parallel mit dem gleichen Qualitätsanspruch bewältigen. Dies hängt insbesondere mit der sehr hohen Komplexität unserer Produkte zusammen. Drachenkrieg enthält beispielsweise mittlerweile weit über 1 Million Wörter an Content.

Zwei Aufgabenbereiche, eine Abteilung

Um eine umfassende Betreuung beider Aspekte zu gewährleisten, übernehmen zwei Abteilungen diese Aufgabe. Während sich das Game-Master-Team, das von einem Product Manager geleitet wird, mit allen In-Game relevanten Aufgaben, wie zum Beispiel der Organisation von Ingame-Events, beschäftigt, übernimmt die Community-Management-Abteilung die Kommunikation mit den Spielern. Sie erfüllt die Rolle eines Sprechers, indem sie bestimmte Produkte, hier in Form von In-Game-Events, an die Community weitergibt. Dieser Vorgang lässt sich jedoch auch umkehren: Das Feedback der Spieler landet zuerst beim Community Manager, der es dann an den Game-Master übergibt. Weil die Position des Community Managers bewusst als erste Anlaufstelle für User gesetzt ist, dient sie somit auch als erste Angriffsinstanz bei Unmut und Unzufriedenheit innerhalb der Community. Die Anordnung der Abteilungen gewährleistet einen hohen Qualitätsstandard der beiden Bereiche. Die Kommunikation mit dem User wird durch qualifizierte Community Manager vorgenommen, die darauf spezialisiert sind, auch in Krisensituationen einen kühlen Kopf zu bewahren. Die Game Master, die meist über einen technischen Background verfügen, füllen indessen die Spiele mit Leben.

Community Management und Product Management bilden gemeinsam die Product-Operations-Abteilung. Für deren Koordination zeichnet sich der Head of Product Operations verantwortlich. Er ist ebenfalls für die Konzeption, Optimierung und Umsetzung der Abläufe und Prozesse zwischen den beiden Subabteilungen zuständig. Der Grund für eine gemeinsame Abteilung ist relativ simpel. Sowohl Product- als auch Community Management verfolgen ein gemeinsames übergeordnetes Ziel – den wirtschaftlichen Erfolg eines Produkts. Um diesen zu erreichen, ist eine enge Zusammenarbeit essentiell und kann nur dann gewährleistet werden, wenn beide Abteilungen so nah wie möglich in der Unternehmensstruktur angesiedelt sind.

Community Management am Beispiel von Drachenkrieg

Das Spiel »Drachenkrieg« wird momentan von fünf internationalen Community Managern betreut, die jeweils eine Sprache abdecken. Das Product-Management-Team besteht aus einem Product Manager, der die Gesamtverantwortung für das Produkt trägt, sowie aus sechs Game-Mastern, die für die In-Game-Aktivitäten zuständig sind. Alles, was im Spiel passiert, wird zunächst der Community mitgeteilt. Erst dann

Beruflich

- Arbeitserfahrung im Bereich Community Management und im Kundensupport, idealerweise in der Spielebranche
- Spielerfahrung mit Multiplayer-Onlinespielen / Browsergames
- Muttersprachler
- Gute Deutsch- und Englischkenntnisse
- Technisches Know how für die Einrichtung und Pflege von Foren- und Kommunikationssystemen
- Besonders wichtig sind Ihre Kommunikationsfähigkeit, Ihre Internetaffinität sowie Ihr ergebnisorientierter und selbstständiger Arbeitsstil.

Persönlich

- extrovertiert
- proaktiv statt reaktiv
- überdurchschnittliches Engagement und Leistungsbereitschaft
- kommunikativ
- Internetaffinität
- ergebnisorientierter und selbstständiger Arbeitsstil
- eigenverantwortliche Arbeitsweise
- soziale Kompetenzen wie die Fähigkeit, Konflikte zu lösen, die Fähigkeit, Mitmenschen zu motivieren und zu führen
- hohe emotionale Intelligenz
- die Fähigkeit, Befindlichkeiten und Strömungen innerhalb der Community zu erkennen

Abbildung 1: Das Beispiel eines **Anforderungsprofils** für einen Community Manager bei Mail.Ru Games.

Abbildung 2: Community- und Product-Management sind in einer **gemeinsamen Abteilung** untergebracht. Nur diese Organisation gewährleistet die notwendige eine enge Zusammenarbeit.

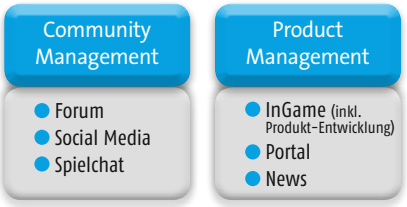


Abbildung 3: Die Aufteilung der **Verantwortungsbereiche** für jedes einzelne Produkt.

wird es auf den Server übertragen. Diese Informationen werden vom Product Management an das Community Management übergeben. Im Gegenzug wird das Feedback der Community vom Community Management an das Product Management übermittelt. Was sich hier so banal anhört, stellt Mail.Ru Games bei der enormen Anzahl an Informationen und Mitarbeitern, die an einem Produkt arbeiten, vor einige Herausforderungen. Wichtige Fragen sind, welche Kanäle sich eigentlich am besten dazu eignen, den Informationsverlust zu minimieren oder sogar zu eliminieren. Und in welcher Beziehung stehen Product- und Community Management überhaupt zueinander?

Prozess- und Kommunikationsstruktur

Um den Kommunikationsweg zwischen beiden Abteilungen kurz zu halten, kann jedes Teammitglied jedes andere jederzeit direkt kontaktieren. Dies geschieht überwiegend per Skype oder E-Mail. E-Mails werden innerhalb der Firma meist zu rein informativen Zwecken genutzt. Es findet zwar auch ein Kommunikationsaustausch statt, grundsätzlich werden aber keine produktbezogenen Aufgaben per Email verschickt. Über Skype werden kleine Angelegenheiten geklärt oder, falls notwendig, größere Aufgaben aufpoliert. Weil dies jedoch wenige Dokumentationsmöglichkeiten und kaum Übersichtlichkeit bietet, eignen sich Skype und Mail nicht als Aufgaben-Tool. Hierfür benutzen wir eine professionelle Projektmanagement-Software. So können wir sicherstellen, dass jede Aufgabe einem oder mehreren Empfängern zugeteilt und eine Deadline gesetzt ist. Dies sorgt für eine klare und übersichtliche Aufgabenzuteilung. Die daraus resultierenden Vorteile überwiegen deutlich die Nachteile, beispielsweise den höheren bürokratischen Aufwand.

Jede Woche finden bei uns außerdem Meetings statt, an denen sowohl Community Manager als auch Game Master eines bestimmten Produkts teilnehmen. Die Meetings sollen den Informationsfluss zwischen den beiden Abteilungen nochmals verbessern.

Ein Auszug aus dem **offiziellen Drachenkrieg-Forum**, dem zentralen Kommunikationskanal der Community, in dem unter anderem die freiwilligen Helfer Guides und andere Artikel erstellen.

Bei dem gesamten Prozedere erfüllt die Community-Management-Abteilung die Rolle eines Dienstleisters. In klaren Worten heißt das: Das Product Management steht in der Pflicht, die Community Manager mit allen notwendigen Informationen zu versorgen, die an die Spieler weitergegeben werden sollen. Um dies umsetzen zu können und Missverständnisse zu vermeiden, müssen klare Verantwortungsbereiche definiert werden (**Abbildung 3**).

Produkt-Teams statt Sprachen-Teams

Innerhalb der Community-Management-Abteilung verläuft die Kommunikation noch direkter ab: Alle Community Manager sitzen in einem Büro, wobei sie nach Produkten und nicht nach Sprachen aufgeteilt sind. Alle Community Manager von Drachenkrieg sitzen also gemeinsam in einer Gruppe. Jeden Morgen findet ein kurzes Stand-Up-Meeting statt, bei dem die wichtigsten Informationen zum vergangenen und kommenden Tag ausgetauscht sowie aktuelle Probleme angesprochen werden. Dieses Meeting hat einen rein informativen Charakter und soll den Gedankenaustausch anregen. So können wir einen Großteil der Probleme innerhalb kürzester Zeit lösen oder zumindest identifizieren. Darüber hinaus werden einmal wöchentlich Community-Management-Meetings organisiert, in denen Mitglieder der unterschiedlichen Teams Strategien diskutieren oder Neuigkeiten aus den Bereichen Community Management und Social Media austauschen. Diese wöchentlichen Meetings dienen vor allem der Weiterbildung und der Schulung der Mitarbeiter, wohingegen die täglichen Meetings eine effiziente, tagesaktuelle Planung anstehender Projekte als Ziel haben. Bei umfangreicheren Aufgaben wird die Kommunikation ähnlich dem Austausch mit dem Product Management gehandhabt: Tasks werden über das Projektmanagement-Tool unter der Vorgabe einer Deadline an eine oder mehrere Personen gestellt.

Ein Heer aus Freiwilligen

Eine der größten Herausforderungen für die Community-Management-Abteilung war die Verlagerung vieler Aufgaben an externe Mitarbeiter. Bei Drachenkrieg stehen uns bis zu 80 freiwillige Helfer zur Verfügung – pro Sprachversion. Diese spielen eine entscheidende Rolle in der Produktbetreuung. Dabei ist die Team-Struktur in vier Bereiche aufgeteilt, die unterschiedliche Aufgaben übernehmen (**Abbildung 4**).

Die einzelnen Bereiche sind, abhängig von Erfahrung und Kompetenz der Helfer, pyramidenförmig angeordnet. So fangen zum Beispiel Neulinge ganz unten an und haben mit der Zeit die Möglichkeit, aufzusteigen.

Neben den Vorteilen, wie der Entlastung des Community Managements, die die externe Verlagerung der Aufgaben mit sich bringt, ergeben sich auch gewisse Nachteile. Freiwillige Helfer sind zwar unabdingbar, sie sind in erster Linie aber immer noch Spieler und keine Angestellten.

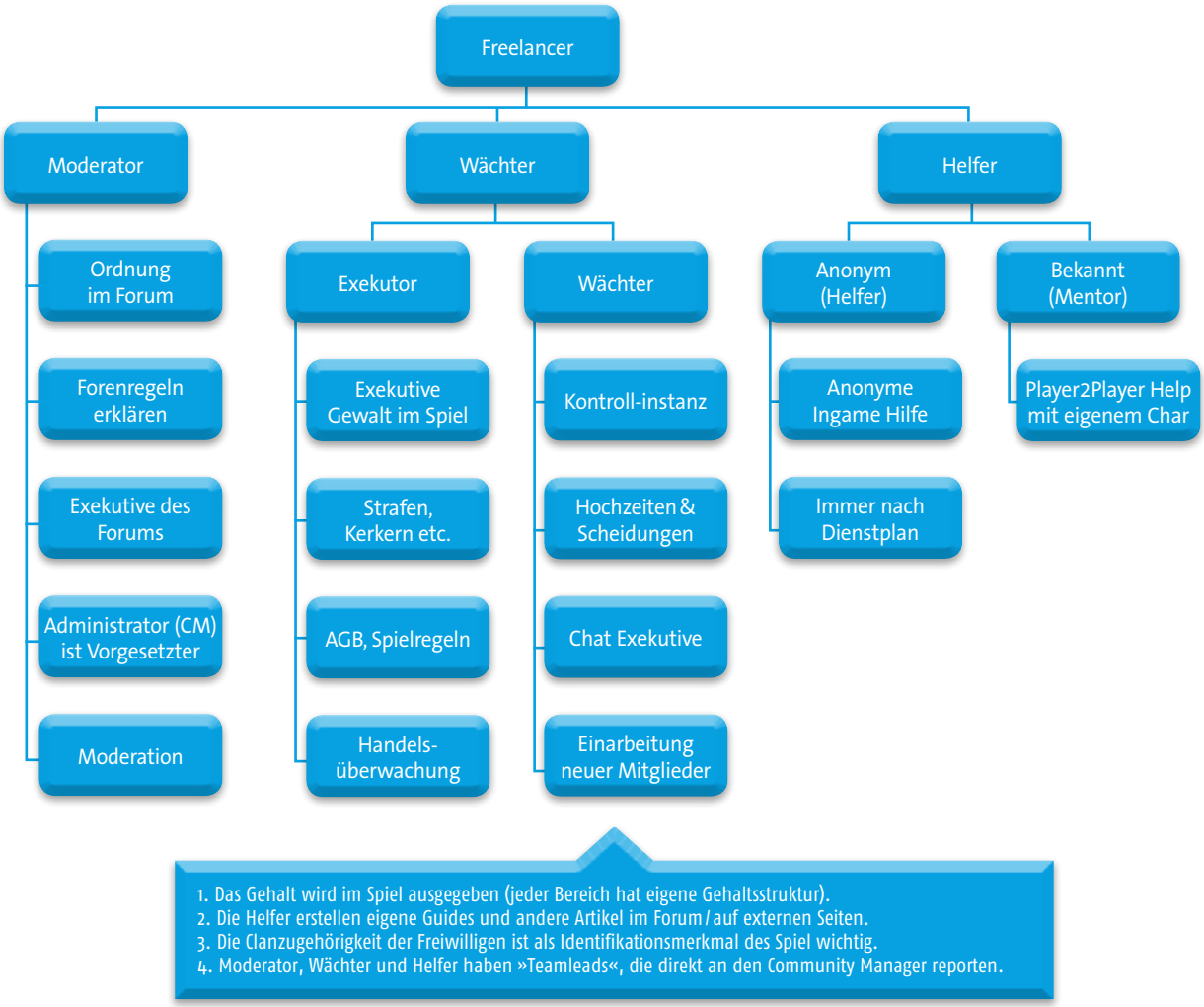


Abbildung 4: Die **Teamstruktur der freiwilligen Helfer** bei Mail.Ru Games. Pro Sprachversion unterstützen 80 Freiwillige das Drachenkrieg-Team.

Deshalb müssen sie sich nicht an gewisse (rechtliche) Vorgaben halten und können für etwaiges Image-schädigendes Verhalten nicht belangt werden. Vom User hingegen werden sie jedoch als offizielle Vertreter des Produkts und somit auch des Unternehmens angesehen und steuern aus diesem Grund wiederum einen erheblichen Anteil zum Image bei. Der Job des Community Managers ist es, die Diskrepanz zwischen der Kompetenz des Freiwilligen als »normalem« Spieler und seiner mangelnden fundierten Kenntnis der Produkte und der firmeninternen Verhaltensmuster zu erkennen und entsprechend zu reagieren. Hierfür werden regelmäßig Schulungen für die Helfer angeboten, in denen exakt diese Aufgaben und Pflichten erläutert werden. Weiterhin muss jeder freiwillige Helfer einen Test bestehen, der seine Helferkompetenz unter Beweis stellt. Dennoch kann nicht ausgeschlossen werden, dass freiwillige Mitarbeiter ihre besonderen Rechte dazu nutzen, im Spiel selbst davon zu profitieren.

Dementsprechend besteht die Notwendigkeit der Kontrolle der freiwilligen Helfer. Hierzu werden bei Mail.Ru Games spezielle Filter und Systeme eingesetzt. Diese automatisieren den Kontrollvorgang und liefern nützliche Informationen, die jedoch vom Community Manager überprüft und ausgewertet werden müssen, weil sie den Vorgang nur erleichtern, jedoch nicht zu 100 Prozent übernehmen können.

Zusammenfassung

Die Notwendigkeit einer Community-Management-Abteilung ist bei vielen Unternehmen aus der Spielebranche bereits angekommen. Diese stehen jedoch oft vor scheinbar unlösbaren Problemen, geht es zum Beispiel um die Suche nach geeigneten Fachkräften. Die Herausforderung besteht darin, das Community Management gezielt in die Unternehmensstruktur einzubauen, was jedoch eine Symbiose diverser Abteilungen und den Einsatz individueller Lösungen erfordert. Auch der Einsatz freiwilliger Helfer muss entsprechend überwacht und koordiniert werden und erfordert einen großen organisatorischen Aufwand und fachliches Know-how innerhalb des Community Managements.

Szymon Wilkosz

So entstand Sims Social für Facebook

Im Gespräch mit Kristian Segerstrale

Wir sprechen mit Playfish-CEO Kristian Segerstrale über die Herausforderung, die altehrwürdige Sims-Marke auf Facebook zu übertragen und welche neuen Impulse dies für die gesamte Social-Games-Landschaft haben kann.

Kristian Segerstrale

ist CEO und Mitgründer von Playfish.

WWW
makinggames.de
- Kristian Segerstrale über Demographie und USPs von Sims Social

> Die meisten Social Games haben nur sehr künstliche und aufgesetzte Formen von Interaktionen.

Making Games Sims Social ist jetzt seit etwa vier Monaten auf Facebook spielbar. Kannst du uns einige Zahlen und Fakten zum Spiel nennen?
Kristian Segerstrale Wir haben mit der Entwicklung von Sims Social ungefähr im August 2010 begonnen. Derzeit befinden wir uns bei knapp 31 Millionen monatlich aktiven Usern (MAUs) und 6 Millionen täglich aktiven Usern (DAUs). Damit ist es aktuell das drittgrößte Spiel auf Facebook.

Making Games Und mit welcher Teamgröße habt ihr entwickelt?
Kristian Segerstrale Gerade bei Social-Games-Projekten variiert die Teamgröße stark. Man fängt mit etwa 8 bis 10 Leuten an und mittlerweile arbeiten mehr als 60 Leute am Spiel. Sims Social ist damit das größte Spiel, das wir je gemacht haben. Im Grunde war es auch genau die Vision, die wir beim Zusammenschluss mit Electronic Arts hatten. Wir wollten von vornherein ein Spiel entwickeln, das völlig natürliche soziale Interaktionen ermöglicht. Wir wollten die Plattform Facebook voranbringen, nicht nur was die Grafik angeht, sondern auch in Bezug auf das Gameplay.
Die meisten Social Games haben nur sehr künstliche und aufgesetzte Formen von Interaktionen.

Mit **Sims Social** transportiert Electronic Arts eine seiner wichtigsten Marken auf Facebook. Der Playfish-Titel ist mit etwa 30 Millionen MAUs derzeit auf Platz 3 der beliebtesten Spiele.

Wir wollten vielmehr eine wirkliche Brücke schlagen zwischen dem Spiel und dem sozialen Umfeld eines Spielers. Dies geschieht zum Beispiel durch die Beziehungen der Sims untereinander. Um dieses Ziel zu erreichen, mussten wir mit mehr Budget in die Entwicklung starten als bei jedem anderen unserer Spiele. In der heutigen Zeit, wenn Spiele als Service angeboten werden, gibt es zwar ein Launch-Budget, die wirkliche Finanzierung entscheidet sich aber nach der Veröffentlichung. Dann kannst du nämlich genau tracken, wie ein Titel ankommt, welche Ressourcen noch nötig sind und entsprechend budgetieren.

Making Games Social Games haben ja typischerweise einen sehr kurzen Development Cycle. Sie gehen früh live und werden dann ständig mit neuen Inhalten versorgt. Verfahrt ihr mit Sims Social genauso oder verfolgt ihr eine andere Content-Update-Strategie? Schließlich sind die Sims-Fans seit Jahren voll ausgestattete und gepolishte Produkte gewöhnt.
Kristian Segerstrale Als vor etwa vier Jahren die ersten Titel auf Facebook live gingen, kam man noch mit wenig Content und unausgereiften Spielen durch. Die Nutzer kannten diese Art von Spielen einfach noch nicht. Seit dieser Zeit hat sich der Markt allerdings rapide gewandelt, eine richtige Evolution der Spiele ist im Gange. Es stimmt also, dass Sims Social bei der Veröffentlichung mehr Inhalte hatte und besser gepolisht war als jedes andere unserer Spiele. Und unsere Roadmap für Updates ist sehr lang. Wenn man bald auf das erste Jahr von Sims Social zurückblickt, wird man feststellen, dass mehr als 50 Prozent des Budgets nach Release ausgegeben wurde.

Making Games Ihr arbeitet eng mit EA Sims zusammen, die alle bisherigen Serienteile entworfen haben. Hat man euch zu Beginn der Entwicklung einen riesigen Katalog mit Must-haves ausgehändigt oder wart ihr völlig frei bei euren Designentscheidungen?
Kristian Segerstrale Die Zusammenarbeit mit EA Sims war von Anfang an hervorragend. Wir waren bei Playfish schon immer riesige Sims-Fans und es gab daher schon eine sehr enge Bindung zur IP. Wir haben zunächst unsere eigene Vision

von einem sozialen Sims-Spiel entworfen. Danach hatten wir diverse Brainstorming-Sessions mit den Leuten von EAs Sims-Studio, was man umsetzen könnte und wie am Ende die IP auf Facebook aussehen sollte. Im gesamten Entwicklungsprozess wurden wir unterstützt, es kamen auch Leute von EA zu uns ins Büro und wurden für eine gewisse Zeit Teil des Teams.

Making Games Hat man euch auch mit dem Code der Original-Spiele oder anderen technischen Grundlagen versorgt?
Kristian Segerstrale Wir mussten Sims Social von Grund auf neu bauen, einfach weil die Plattform es nicht anders hergegeben hätte. Technisch gesehen unterscheidet sich das Spiel sehr von seinen »Vorgängern«. Es existiert somit zwar kein »Shared Code«, dafür aber in einigen Fällen »Shared Audio« oder »Shared Art«. Uns war sehr wichtig, dass das Spiel der Serie treu bleibt, es musste also eine Konsistenz erkennbar sein. Es gab so gut wie keine Stellen, an denen wir einfach etwas via Copy & Paste einbauen konnten, dafür aber Unmengen an Dingen, die von den anderen Sims-Spielen inspiriert sind und in gewisser Weise Übersetzungen darstellen.

Making Games Wie wird sich die Zusammenarbeit in den nächsten Monaten gestalten? Werdet ihr auch weiterhin Teile des Game Designs oder vielleicht auch des Businessmodells teilen?
Kristian Segerstrale Ich gehe davon aus, dass unsere Zusammenarbeit genau so weitergehen wird. Für uns ist ja auch wichtig, dass einige unserer Learnings wieder zurück transportiert werden. Das Ganze ist ziemlich spannend, weil wir uns in der glücklichen Position befinden, Wege auszukundschaften und Plattformen zu besetzen, auf der die Marke Sims noch nie zu finden war.

Making Games Für die Marke Sims ist es ja auch etwas sehr Neues, so viele Daten über das Spiel tracken zu können. Gab es denn überraschende Erkenntnisse, die ihr nun dem Sims-Team von EA mit auf den Weg geben könnt?
Kristian Segerstrale Die Einblicke, die wir mit Sims Social erhalten, sind vor allem auf dem Gebiet sozialer Interaktion spannend. Wir lernen eine Menge darüber, wie das Miteinander realer Personen in einem Spiel funktioniert. Und die Erkenntnisse über diese Dimension fließen dann natürlich wieder ein in die Weiterentwicklung der klassischen Marke auf anderen Plattformen.

Making Games Gibt es etwas, das ihr mit Sims Social bewusst anders macht als alle anderen Spiele auf Facebook?
Kristian Segerstrale Auffällig ist, dass die meisten Social Games dem Spieler einen äußerst begrenzten Energievorrat geben. Man kann dann vielleicht gerade einmal fünf Minuten am Stück spielen, dann sind alle Aktionspunkte verbraucht. Wir wollten mit Sims Social hingegen ein Spiel designen, das man wirklich sehr lange am Stück spielen kann und in das man auch emotional viel investiert. Auch wir haben ein Energiesystem integriert, aber wir wollen dennoch unter allen Umständen verhindern, dass sich unsere Nutzer in ihren Möglichkeiten eingeschränkt fühlen. Ich habe nicht die genaue Zahl im Kopf, aber im Moment spielen unsere Nutzer ungefähr 30 bis 40 Minuten pro Tag.

Making Games Werdet ihr euren Ansatz, den Spieler nicht künstlich einzuschränken, auch in weiteren Spielen verfolgen oder ist dies ein Businessmodell, das nur durch die Mehrdimensionalität im Sims-Universum funktioniert?
Kristian Segerstrale Social Games stecken zwar nicht mehr in den Kinderschuhen, sie sind aber gerade vielleicht mal Teenager (lacht).
Wir entwickeln uns derzeit sehr schnell in der Hinsicht, die Philosophien und Frameworks des Social Gamings zu verstehen. Und dazu gehört auch, dem Spieler Freiraum zu geben, ihn nicht einzuschränken und trotzdem ein funktionierendes Businessmodell zu besitzen. Wir sind sehr stolz darauf, dass wir wirklich bedeutende soziale Interkationen zwischen den Facebook-Nutzern fördern konnten. So werden etwa die Feedposts sehr häufig kommentiert, weil sie einfach witzig sind und etwas über den Spieler aussagen. Und sie werden zu einem Teil der ganz normalen und natürlichen sozialen Interaktionen mit Freunden.

Making Games Gibt es eine Essenz, bestimmte Grundpfeiler, die ein Playfish Game haben muss?
Kristian Segerstrale Unsere Vision bzw. Vorstellung ist, dass die Menschen Social Games nicht vordringlich des reinen Spiels wegen spielen, sondern um mit ihren Freunden zu interagieren. In »Biggest Brain«, einem unserer ersten Spiele, ging es eigentlich nicht um das, was auf dem Spielfeld geschah. Es ging vordringlich darum, wer denn nun das größte Gehirn hat. Bis heute hat eine Evolution stattgefunden, die viel komplexere Interaktionen ermöglicht und der wir mit unseren kommenden Spielen Rechnung tragen wollen. Im Grunde wollen wir mit unseren Spielen nur wieder das etwas verloren gegangene Gefühl einfangen, an einem Sonntag gemeinsam mit Freunden ein Brettspiel zu spielen.

Making Games Wie siehst du grundsätzlich die Zukunft von Social Games? Wo wird die Industrie in ein paar Jahren stehen?
Kristian Segerstrale Ich denke, dass vermutlich schon in drei Jahren niemand mehr von Social Games reden wird. Genauso wenig wie man heute über »Internet Games« oder »Electricity Games« spricht. Es wird nur noch von »Games« die Rede sein und die soziale Sparte dieser Games wird ein fundamentaler Baustein der Landschaft der Computerspiele sein.

Darüber hinaus ich bin fest davon überzeugt, dass sich diese Art von Spielen alle Plattformen erstrecken wird, angefangen beim Browser über Konsolen bis hin zu Smartphones.

Making Games Vielen Dank, Kristian!
Interview: Yassin Chakhchoukh

Eines der ersten Spiele von Playfish war **Biggest Brain**. Laut Kristian Segerstrale ging es darin weniger um den Spielablauf selbst als um den Vergleich mit Freunden. Die Komplexität der sozialen Interaktion in Social Games hat seitdem immer mehr zugenommen.

> Im Moment spielen unsere Nutzer ungefähr 30 bis 40 Minuten pro Tag.

The Sims Social Runs On Dunkin'! TM
Check out the exclusive content available on the Dunkin' Donuts Fanpage!
DUNKIN' DONUTS
OKAY

Playfish arbeitet wie Electronic Arts zuvor schon beim klassischen Die Sims mit Firmen wie **Dunkin' Donuts** zusammen und konzipiert Marketingkampagnen mit speziellen Möbeln oder Verbrauchsgegenständen.

> Wenn man bald auf das erste Jahr von Sims Social zurückblickt, wird man feststellen, dass mehr als 50 Prozent des Budgets nach Release ausgegeben wurde.

Business Analytics und Game Design

Tracking von Mobile Games

So klappt die Zusammenarbeit zwischen Kreativen und Analysten: Die Experten von Remote Control Productions und Chimera Entertainment erklären, mit welchen Tools, Methoden und Prozessen sie die Kennzahlen ihrer Mobile Games tracken und auswerten.

Dennis Raspotnig

ist Business Analyst bei Remote Control Productions.

Dennis hat an der LMU München Neuere Deutsche Literatur, Medienrecht und Psychologie studiert. Seit einem Jahr beschäftigt er sich bei Remote Control Productions mit der Analyse und der Aufbereitung von Kennzahlen aus dem Mobile- und Online-Bereich. Seine aktuellen Projekte sind das Browserspiel Warstory – Europe in Flames sowie die Smartphonegames Demolition Dash und Happy Hills.

Rene Schölzel

ist Programmierer bei Chimera Entertainment.

Rene hat von 2006 bis 2008 als Freelancer vorrangig für Versicherungsunternehmen gearbeitet, in dieser Zeit hat er zum Fachinformatiker umgeschult. Seit 2008 arbeitet er beim Münchner Entwicklerstudio Chimera Entertainment mit dem Schwerpunkt Unity Coding für Mobile und Online. Seine bisherigen Spieleprojekte sind Gehirntraining mit Dr. Kawashima, Farm Fieber/Wiesn Fieber, Ravensburger memory®, Warstory – Europe in Flames, Demolition Dash und Happy Hills.

Die Entwicklung und Erfolgskontrolle durch das gezielte Auswerten von Daten gehört spätestens seit dem Aufschwung von Social-Gaming-Firmen wie Zynga oder Kabam auch bei der Entwicklung von Mobile Games zum technischen und strategischen Standard. Aber an welchen Stellen können sich die Business Analytics sinnvoll in den Entwicklungsprozess einbringen? Was ist die perfekte Balance zwischen klassischen kreativen Prozessen des Game Designs und zahlengetriebener Performance-Optimierung der Business Analytics?

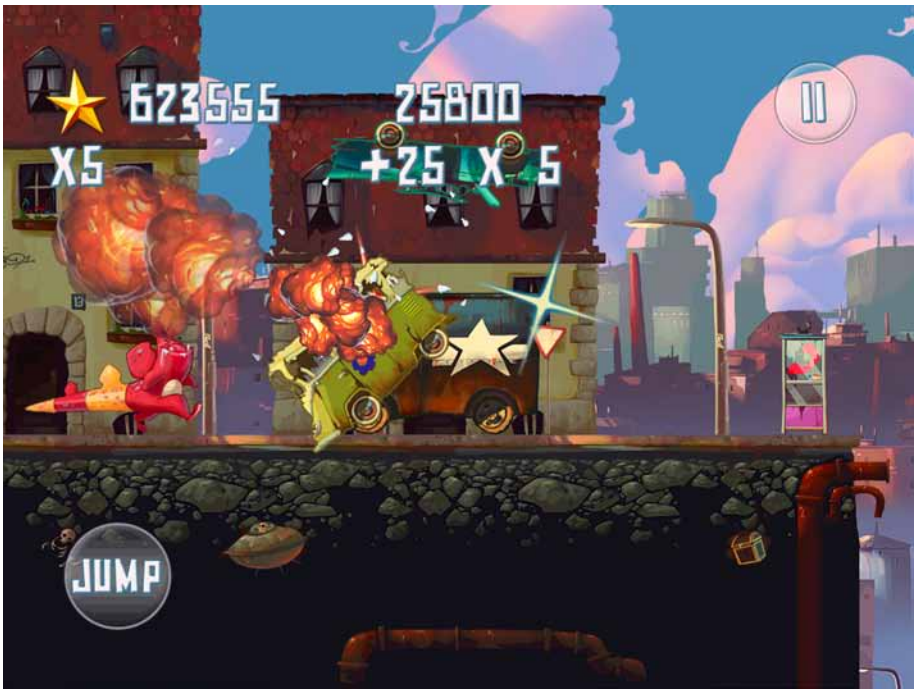
Die Konzeptphase

Grundsätzlich lassen sich zwei große Phasen der Entwicklung benennen, bei denen Business Analytics eine entscheidende Rolle spielen können: die Konzept- sowie die Optimierungsphase.

Am Anfang der Entwicklung stehen Idee und Ausarbeitung des Grobkonzepts. Schon hier ist es sinnvoll, eine enge Zusammenarbeit zwischen Game Design und Business Analytics anzustreben. Obwohl es am Produkt noch nichts Konkretes zu messen gibt, sollte man bereits gezielte Recherche-Anfragen an das Business-Analytics-Team stellen, um kreativ-ferne Fragen rechtzeitig in das Design einbeziehen zu können. Vor allem Entscheidungen bezüglich der Preisbildung werden getroffen: Wie soll mein Produkt monetarisieren? Welchen Umsatz machen andere Spiele aus diesem Genre? Wie ist die Wettbewerbssituation? Wie teuer sind ähnliche Apps?

Dabei wird das vorhandene Grobkonzept des Game Designs mit bereits bestehenden Konkurrenzdaten verglichen. Sinnvolle Anlaufstellen sind etwa Anbieter wie Xyologic (www.xyologic.com) oder App Annie (www.appannie.com), die Daten kostenlos zu Verfügung stellen.

Die Erkenntnisse zu Wachstumsraten im angepeilten Markt, zur Konkurrenzanalyse und zu Monetarisierungsformen sollten bereits in die frühen Iterationen des Game Designs einfließen. So lassen sich potenzielle Fehler vermeiden und »Best Practices« adaptieren. Es lohnt sich, viel Aufwand in diese Phase zu stecken. Im Gegensatz zu Browserspielen dauert es aufgrund der iTunes-Geschäftsbedingungen deutlich länger, Updates einzuspielen. Zudem muss der



Für den Erfolg eines Smartphone-Spiels wie **Demolition Dash** müssen Game Design und Business Analytics Hand in Hand gehen. Nur so können Fehlerstellen effizient aufgespürt und behoben werden.

Spieler das Update nicht zwingend annehmen. Je nach Detailgrad des Designs werden auch die Fragen an das Business-Analytics-Team detaillierter: Welches Businessmodell ist sinnvoll? **Abbildung 1** bietet Ihnen eine Übersicht der gängigen Businessmodelle im Mobile-Bereich.

Welche Art von In-App Purchases monetarisieren im angestrebten Genre bisher am besten? Und was ist der durchschnittliche Preis des meistgekauften »In-App Purchase« im Genre? Und lässt er sich einer Kategorie zuordnen? Nachdem diese Fragen geklärt sind und das Design steht, startet der Entwicklungsprozess. Hier beginnen aus Sicht der Business Analytics die eigentliche Erfolgskontrolle und der damit verbundene Optimierungsprozess.

Die Optimierungsphase

Zuerst muss geklärt werden, welche Daten wie erfasst werden sollen. Das »Wie« ist aus technischer Sicht einfach zu beantworten, weil es nur zwei grundlegende Ansätze gibt: Nutze ich fertige oder individuelle Plugins zur Datenerhebung? Oder anders ausgedrückt: Möchte ich Plugins von etablierten Tracking-Tools verwenden oder gehe ich sogar so weit, selbst gebaute Plugins in den Code zu implementieren?

Natürlich ist es einfacher, auf vorhandene Plugins zurückzugreifen, allerdings hat dies auch Nachteile. Metriken sind sensible Daten, die man ungern an Dritte herausgibt. Außerdem ist eine vorgefertigte Lösung unflexibler als eine an das Spiel angepasste. Aufgrund dessen wählt man meistens einen Kompromiss und integriert sowohl fertige als auch individuell angepasste Plugins.

Bei der Wahl der Third-Party-Software spielt eine entscheidende Rolle, was denn eigentlich gemessen werden soll. Interessante Daten können zum Beispiel sein: Anzahl neuer Spieler, Anzahl aktiver Spieler, Anzahl Sessions, Session-Länge, Spielerherkunft usw.

Dabei sollten möglichst alle Daten einfach zueinander in Bezug gesetzt werden können. In **Abbildung 2** finden Sie eine kurze Zusammenfassung von interessanten Plugins. Wichtig ist vor allem, dass auch für den mobilen Bereich die »90/10«-Regel von Avinash Kaushik, dem Autor von »Web Analytics: An Hour A Day«, gilt: Investiere ich 10 Euro in entsprechende Software, sollte ich mindestens 90 Euro in gute Business-Analysten investieren. Daten sind nur dann wertvoll, wenn es jemanden gibt, der sie gewinnbringend auswerten kann.

Ein etablierter Softwareanbieter zur Datenerhebung von Spielerverhalten ist Flurry. Die Implementierung gestaltet sich relativ einfach. Um sie aus technischer Sicht besser darstellen zu können, hat sich Rene Schölzel, Unity Coder bei Chimera Entertainment, bereit erklärt, die Integration von Flurry in eine bestehende Xcode-Anwendung zu beschreiben. Rene hat für Chimera Entertainment bereits fünf iOS-Projekte auf Unity-Basis entwickelt, darunter Demolition Dash, Happy Hills und Ravensburger memory®.

Paid Download
Die klassische Variante des E-Commerce. Die App wird gegen Bezahlung aus dem App Store geladen und bietet dem Nutzer keine weiteren Möglichkeiten, im Spiel Geld zu investieren.

Freemium
In diesem Businessmodell wird das Spiel gratis zum Download bereitgestellt. Umsatz wird durch Zusatzangebote generiert, den sogenannten »In-App Purchases«. Das können etwa Verbrauchsgüter zu Verlängerung der Spielsession, Powerups für den Kampf mit anderen Spielern oder zusätzliche Spielinhalte sein. Auch Abo-Modelle sind möglich, allerdings unterliegen diese derselben Logik wie andere In-App Purchases.

PD + In-App Purchases
Eine Mischvariante aus den beiden genannten Businessmodellen. Häufig werden Freemium-Modelle auch gegen Bezahlung angeboten, weil die Top-Platzierungen der meistgekauften Spiele stabiler sind und sich so zusätzlicher Umsatz erwirtschaften lässt.

Abbildung 1: Ein Überblick über die gängigsten **Businessmodelle** im Mobile-Bereich.

Flurry-Integration in Xcode-Anwendung

Die Integration von Flurry Analytics in eine bestehende Xcode-Anwendung ist einfach und kann innerhalb weniger Minuten abgeschlossen werden. Als Erstes müssen Sie einen Account bei Flurry (www.flurry.com) erstellen. Für die erste Applikation gehen Sie auf den Reiter »Applications« in der Navigationsleiste und wählen »Add a New Application«. Dort wird das entsprechende SDK zum Download ausgewählt, in diesem Fall das iPhone-SDK bzw. das iPad-SDK.

Als Nächstes werden Namen und Kategorie der Anwendung gewählt. Sind beide Felder ausgefüllt und die Anwendung erstellt, landet man direkt auf der Integrationsseite der erstellten Anwendung. Ganz oben findet sich der Download-Link zum Flurry-SDK, um das Archiv herunterzuladen. Anschließend kopieren Sie die entpackten Dateien in das Projektverzeichnis und fügen sie dem Projekt in der Xcode-IDE hinzu, indem Sie über das Project-Menü »Add to project« und »FlurryAnalytics« wählen.

Bei den Import-Optionen wählen Sie »Recursively create groups for any added folders« aus. Nachdem Sie das SDK erfolgreich in das Projekt integriert haben, müssen Sie nur noch einige Zeilen Code hinzufügen. Dafür öffnen Sie die AppController-Datei und fügen in den Header-Dateien folgende Zeilen ein:

```
#import „FlurryAnalytics.h“
```

Danach suchen Sie nach dieser Zeile

```
(void)applicationDidFinishLaunching:(UIApplication *)
application
```

und fügen Folgendes ein:

```
[FlurryAnalytics startSession:@“XXXXXXXXXXXXXXXX“];
```

Wobei XXXXXXXXXXXXX für den Flurry Application Key steht, den man auf der Integrationsseite ganz oben findet. Der Key ist außerdem auch unter »Punkt 2 Integration« zu finden. Damit haben Sie die Basis-Integration von Flurry Analytics abgeschlossen.

Daneben ist es sinnvoll, eigene Custom Events zu erstellen, um individuelle Spielabschnitte zu messen, die mit der Basis-Integration nicht abgebildet werden können. Außerdem empfiehlt es sich, den Error Handler für unerwartete Fehler

Scoreloop
Bietet ein Crossplattform-Ökosystem für Spiele.

Appirater
Erinnert Ihre Nutzer daran, das Spiel im iTunes Store zu bewerten.

Flurry
Trackt das Spielerverhalten.

StoreKit
Framework von Apple, um einen In-App Store zu implementieren.

Tapjoy
Schaltet Werbung und generiert neue User.

Admob
Schaltet In-App-Werbung.

Apprupt
Ermöglicht Mobile Advertising.

Abbildung 2: Eine Auswahl **nützlicher Plugins** für die Entwicklung Ihres Smartphone-Spiels.

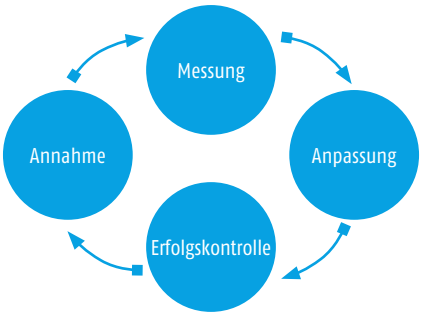


Abbildung 3: Der Ablauf der **Optimierung** ist zyklisch aufgebaut und erfolgt daher immer gleich.

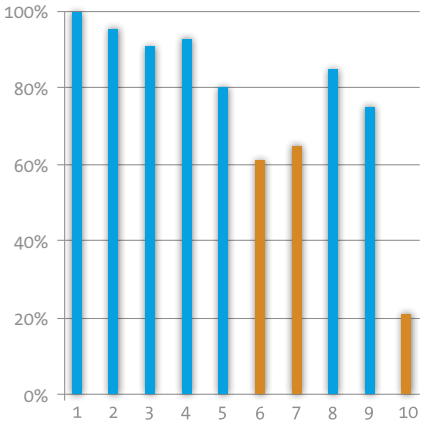


Abbildung 4: Analytics-Beispiel **Tutorial** – hier wird deutlich, welche Einzelschritte von den Spielern mit welcher Häufigkeit abgeschlossen werden.

hinzuzufügen. Das hilft bei der Weiterentwicklung von Flurry und sorgt für ein besseres Produkt. Um den Error Handler zu integrieren, erstellen Sie folgende Funktion in der Anwendung:

```
void uncaughtExceptionHandler(NSException *exception)
{
   [FlurryAnalytics logError:@"Uncaught"
message:@"Crash!" exception:exception];
}
```

Diese Funktion schickt beim nächsten Start der Anwendung eine Meldung an Flurry, sollten Fehler auftreten. Damit die Funktion im Fehlerfall aufgerufen wird, muss man sie noch bekannt machen und registrieren. Dazu kehren Sie wieder zu der folgenden Funktion zurück

```
(void)applicationDidFinishLaunching:(UIApplication *)
application
```

und fügen in der ersten Zeile diesen Ausdruck ein:

```
NSSetUncaughtExceptionHandler(&uncaughtException
Handler);
```

Damit wurde auch das Error Handling erfolgreich integriert. Sollten Sie keine weiteren benutzerdefinierten Events hinzufügen, ist die Integration abgeschlossen. Die erfolgreiche Integration lässt sich einfach testen, indem Sie einen Development Build auf ein Gerät aufspielen und starten. Die Flurry-Daten sollten innerhalb einer Stunde aktualisiert sein.

Sobald die Implementierung abgeschlossen ist, beginnt die Optimierungsphase, die immer ein grundlegendes Ziel haben sollte: Wie kann ich mein Spiel verbessern, um meine Nutzer zufriedenzustellen? Zufriedene Kunden investieren mehr Zeit und, wenn es das Businessmodell zulässt, auch mehr Geld. Zufriedene Kunden bewerten mein Spiel darüber hinaus auch besser und empfehlen es ihren Freunden.

Mit dieser Frage im Hinterkopf beginnt die Suche nach kritischen Stellen, die die Spieler nicht mögen. Die Vorgehensweise gestaltet sich grundsätzlich immer gleich und lässt sich am besten in einem Kreislauf darstellen (**Abbildung 3**).

Richtige Fragen, verwertbare Antworten

Zuerst gilt es, eine fundierte Annahme zu treffen oder eine spielspezifische Frage zu stellen. Diese kann sich aus bereits geschehenen Auswertungen ergeben, aber auch aus dem Feedback von Spielern im iTunes Store, in Foren oder auf Fanseiten. Häufig sind der Spieleinstieg und das Tutorial (sofern vorhanden) eine kritische Stelle. Eine Frage könnte also sein: Beenden die Spieler mein Tutorial oder brechen sie es frühzeitig ab?

Im zweiten Schritt gilt es nun, diese Annahme mit Messungen zu überprüfen: **Abbildung 4** zeigt beispielhaft die einzelnen Tutorial-Schritte und mit welcher Häufigkeit sie von den Spielern abgeschlossen werden. Sofort wird deutlich, dass sie mit Schritt 6, 7 und insbesondere mit Schritt 10 unzufrieden sind. Diese Problemstellen müssen also so schnell wie möglich vom Entwickler ausgebessert werden.

Die Anpassung ist meistens der schwierigste Teil des Prozesses, weil es dafür keine harten Daten gibt. Hier müssen die Game Designer ran, um herauszufinden, warum diese Schritte auf den Spieler unbefriedigend wirken. Vor allem Erfahrung durch »Trial and Error« und ein guter Überblick, wie erfolgreiche Konkurrenzprodukte ihr Tutorial gestalten, können zum Ziel führen. Es lohnt sich gegebenenfalls auch ein Fokusgruppentest, sollte dieser zeitlich möglich sein und das Budget nicht sprengen.

Kein Erfolg ohne Erfolgskontrolle

Sobald alle Anpassungen implementiert und in den iTunes Store geladen wurden, geht es an die Erfolgskontrolle. Wenn sich die Zahlen wie erhofft verbessert haben, beginnt der ganze Optimierungsprozess von vorn, es muss eine weitere Annahme getroffen werden usw.

Bei der Anpassung des Spiels zeigt sich, wie wichtig eine enge Zusammenarbeit zwischen Game Design und Business Analytics ist. Auch wenn es nicht immer einfach ist, kreative Gedanken und trockene Zahlen unter einen Hut zu bringen, lohnt es sich, frühzeitig klare Arbeitsprozesse zu definieren. Nur durch eine gute Zusammenarbeit lässt sich das volle Potenzial der Business Analytics ausschöpfen. Zudem wird klar, dass gute Analysearbeit allein kein Spiel zum Hit macht. Der Erfolg eines Spiels steht und fällt mit seinem Design. Trotzdem lässt sich durch das Auswerten und Kontrollieren von Spieldaten ein enormer Mehrwert schaffen.

Man sollte sich also nicht die Frage stellen, ob es besser wäre, sich auf die Erfahrung der Game Designer oder doch lieber auf analytische Daten zu verlassen. Stattdessen sollte das Credo lauten: Wie kann ich die Zusammenarbeit zwischen Game Design und Business Analytics möglichst effizient gestalten und stärker forcieren?

Dennis Raspotnig und Rene Schölzel

Entwicklungsverträge von Online Games, Teil 3
Mitwirkungsleistungen & Produktpflege

Oft gehört: Onlinespiele sind keine Produkte, sondern ein Service. Entsprechend muss vertraglich geklärt werden, wer für welche Serviceleistungen verantwortlich ist und wie dies vergütet wird.

Die Mitwirkungs- und Beistellleistungen des Publishers / Plattformbetreibers sind ein wichtiger Teil der Entwicklung. Sie sind erforderlich, um das Spiel in die Plattform zu integrieren und damit natürlich auch essenziell für ein erfolgreiches Produkt. Viele Verträge behandeln diesen Punkt zu knapp: Sie erschöpfen sich bestenfalls in der Verpflichtung des Publishers, die notwendigen Unterlagen, Daten und sonstige Informationen bereitzustellen. Ebenso verpflichtet sich der Plattformbetreiber zu sonstigen allgemeinen Mitwirkungsleistungen, die für die Erfüllung der Leistungspflichten des Entwicklers erforderlich sind. Wenn im Konfliktfall der Entwickler hieraus einen konkreten Anspruch oder eine Einwendung herleiten will – zum Beispiel indem er sich für die verspätete Fertigstellung damit entschuldigt, dass der Publisher/Plattformbetreiber nicht alle erforderlichen Spezifikationen für die Anbindung des Spiels geliefert hat – dann muss er selbst darlegen (und beweisen), dass (i) die Notwendigkeit für diese konkrete Mitwirkungsleistung bestand und (ii) dass der Publisher diese auch liefern konnte. Klarstellen sollte der Vertrag auch, ob es sich bei der Mitwirkung um echte Leistungspflichten handelt (dann kommt auch der Publisher in Verzug und muss ggf. den entstandenen Schaden ersetzen) oder ob es sich um bloße Obliegenheiten handelt.

Mitwirkungsleistungen definieren

Nicht selten entstehen Verzögerungen durch Fehler des Publishers bei der Erfüllung seiner Mitwirkungs- und Beistellleistungen. Für den Projekterfolg und den pünktlichen Launch kann dadurch eine kritische Situation entstehen, ohne dass den Entwickler die Verantwortung trifft. Dabei ist dem Publisher oft nicht bewusst, dass seine Mitwirkung erforderlich ist. Wichtig sind deshalb möglichst konkrete Aufzählungen der zu erbringenden Mitwirkungsleistungen. In vielen Fällen wird der Publisher eine Plattform bereitstellen müssen sowie Hardware und Server oder eine Cloud-Lösung stellen. Die Definition der Umgebung und deren Leistungsfähigkeit gehören ebenfalls in den Vertrag. Beispielsweise unterscheiden sich die Anforderungen bei der Integration von Client-basierten-Spielen in eine Plattform deutlich von den Anforderungen an die Integration von Browserspielen. Der Vertrag sollte hierzu eine Definition der Plattform, des Hardware-Sizing oder einen Konfigurationsvorschlag enthalten. Die Zeitpunkte, wann diese zur Verfügung gestellt werden, gehören ebenfalls in die Leistungsbeschreibung. Während eine Entwicklungs- oder Testumgebung bereits bei der Entwicklung benötigt wird, genügt es, wenn die eigentliche Spielumgebung bei der Migration zur »Closed Beta« bereitsteht. Die Einbindung von Bestellprozessen beginnt erst mit der »Open Beta«, spätestens aber mit dem Live-Betrieb des Spiels. Sollen die Spiele auf proprietären Umgebungen ablaufen, sind die besonderen Vertragsbedingungen zu integrieren. Das gilt für etwaige SDK's, aber auch für die Plattformbedingungen von Facebook etc.

Für die Vertragsgestaltung bietet es sich an, diese Bedingungen als Anlage zu dem Vertrag zu nehmen und in den Definitionen einzubeziehen. Solche Anlagen (eigenständige Dokumente) werden traditionell als »Exhibits« bezeichnet. Im Gegensatz dazu sind »Schedules« Bestandteile eines Vertrages, die aus Platz- oder Praktikabilitätsgründen aus dem eigentlichen Vertragstext in eine Anlage separiert werden. Üblicherweise fallen darunter Milestones, Preislisten, technische Beschreibungen, Glossare etc.

Produktpflege

Für den Erfolg von Onlinespielen ist das ständige Anpassen, Verändern und Eingehen auf die Wünsche der Community unerlässlich. Das Einbringen neuer Inhalte, Aufgaben, Features etc. hält Kunden bei der Stange und ermöglicht die bessere Auswertung des Spiels. Das gilt heute umso mehr, als die Gewinnung neuer Spieler immer kostspieliger wird und die Kundenbindung an Bedeutung gewinnt. Der Publisher besteht deshalb in der Regel darauf, dass der Entwickler weitergehende Leistungen erbringt als nur Fehlerbeseitigungen.

Claas Oehler

ist Rechtsanwalt bei IHDE & Partner.

Claas Oehler hat in Berlin Musikwissenschaften und Rechtswissenschaften studiert. Als Rechtsanwalt ist er seit 2002 zugelassen. Er war wissenschaftlicher Mitarbeiter an der Europa-Universität Viadrina (Lehrstuhl für Wirtschaftsrecht). Seit 2002 ist er in Berlin als Rechtsanwalt tätig, bei der Kanzlei IHDE & Partner seit 2006.

Inhalte der Artikelreihe

I. Einführung
Entwicklungs- und Betriebsverträge für Onlinespiele
1. Von woher kommt das Spiel auf die Plattform?
2. Vertragsarten und Abgrenzung
3. Auftragsproduktion, Besonderheiten beim Entwicklervertrag für Onlinespiele
4. Andere Bereitstellungsmodelle B2B: White Label, Software as a Service, Applikation Service Providing, Lizenzmodelle

II. Wie wird das Spiel für den User bereitgestellt? Verträge
1. Hostingvertrag
2. Webspace, Server Housing
3. Betreibervertrag, Wartung, Pflege
4. Reichweitenpartner
5. Payment Provider
6. Plattformbedingungen, Social Games

III. Rahmenbedingungen
1. Verträge mit Spielern, ToU, AGB
2. Einführung: AGB, Jugendschutz, Werberecht, Verbraucherschutz
3. Virtuelle Güter, Plattformen
4. Betreiberpflichten
5. Die fünf größten Risiken bei Browserspielen

Die Vergütung für diese Weiterentwicklung wird der Entwickler zu seinen Stundensätzen/Manntagessätzen abrechnen wollen. Um sicherzustellen, dass solche Leistungen überhaupt erbracht werden, sollte der Vertrag regeln, wie lange die Pflicht zur Weiterentwicklung und Anpassung besteht und in welcher Höhe sie zu vergüten ist. Viele Verträge versuchen das mit einer Pauschale abzudecken, die monatliche Fixbeträge beinhalten und dann mit Lizenzgebühren verrechnet werden. In der Sache sind die Klauseln mit den »Change Request«- bzw. »Change Management«-Klauseln verwandt, sodass auf diesen Beitrag verwiesen werden kann.

Üblich ist es auch, die sonstigen Kosten zu vergüten, die dem Entwickler durch die Pflegeleistungen beim Publisher vor Ort entstehen, ebenso die damit verbundenen Reisekosten. Aus diesem Grund wird der Publisher ein vitales Interesse daran haben, dass diese Kosten nur dann zusätzlich vergütet werden, wenn eine »Remote Administration« scheitert, weil diese in der Sache nicht möglich ist oder dies auf Gründen beruht, die der Publisher zu vertreten hat. Die Höhe der zusätzlichen Kosten richtet sich in der Regel nach den Entwicklersätzen, die ebenfalls im Vertrag geregelt sein und die sich jährlich anpassen sollten.

Klauseln für die Produktpflege sollten regeln, welche Vergütung für die Produktpflege geleistet wird und wie lange die Produktpflege verbindlich ist. Welche Leistungen fallen unter die Produktpflege und welche Leistungen sind bereits mit der Lizenzvergütung abgegolten? Was ist mit Sicherheitslücken, Einbindung neuer Middleware und Zahlungsvarianten?

Üblich ist, dass der Entwickler während der Betriebszeit eine Pflegeverpflichtung übernimmt. Dafür erhält er wiederum eine feste Vergütung, einen Anteil an den Einnahmen oder eine Mischvergütung. Häufig sind Umsatzbeteiligungen an den laufenden Erträgen des Onlinespiels. Diese können einerseits als Lizenzgebühren angesehen werden, zum anderen aber auch Vergütungen für Zusatzleistungen sein. Andere Modelle regeln eine fixe Wartungspauschale, die dann aber von den Umsatzbeteiligungen abgezogen wird. Der Vertrag sollte aber sorgsam abgrenzen, welche Vergütung für welche Leistung gezahlt wird. Sonst können rechtliche Probleme auftreten, die ich im Folgenden näher beschreiben werde.

Problem 1: Doppelerfassung der Gewährleistung

Bei einer dauerhaften Lizenzierung und einmaligen Vergütung für die Erstellung des Spiels wird ein Kauf oder Werk- oder Werkliefervertrag vorliegen. Der Entwickler ist grundsätzlich zur Mangelbeseitigung auf seine Kosten verpflichtet. Wird das Spiel nur auf Zeit (etwa fünf Jahre für das europäische Territorium) lizenziert, so wird man das rechtlich als Pacht oder Miete einordnen müssen. Bei solchen Verträgen wäre es aber genau die Aufgabe des Entwicklers, das Spiel in einwandfreiem Zustand zu erhalten. Was Gewährleistung ist, wird oft über die Definition des Mangels (defect/error/bug) innerhalb des Vertrages (mit-)gesteuert. Allgemein gilt als Mangel des Spiels jede Abweichung der Funktionalität und Qualität von den Beschreibungen im Vertrag (wenn nichts beschrieben ist) von der mittleren Art und Güte. Der Entwickler wird ein erhebliches Interesse daran haben, nur solche Probleme des Spiels im Rahmen der Pflegeleistungen zu beheben, die auf Mängeln des Spiels, also letztlich auf ein eigenes Verschulden des Entwicklers während der Erstellungsphase zurückzuführen sind.

Die Leistungen, die der Entwickler ohnehin zu erbringen (Fehlerbeseitigung, Patches etc.) hat, und neue Leistungen (Erweiterung des Spiels, neue Feature, Sprachen, Anpassung an neue Browserversionen, Treiber etc.) überschneiden sich. Im Ergebnis führt das zu einer Doppelvergütung, wenn nicht sauber abgegrenzt wird. In Formularverträgen kann eine derartige Doppelvergütung zu Wirksamkeitsproblemen bei bestimmten Klauseln führen.

Problem 2: Steuerrechtliche Zuordnung

Die Vergütung, die als Gegenleistung für das Überlassen des Spiels gezahlt wird, lässt sich auch als Lizenzgebühr einordnen. Diese Lizenzgebühren unterscheiden sich steuerrechtlich allerdings von den Vergütungen für Wartungsleistungen. Zum einen unterliegen die Zahlungen für das Einräumen von Rechten nach dem Urhebergesetz dem ermäßigten Umsatzsteuersatz, zum anderen können, bei grenzüberschreitenden Konstellationen sogenannte Quellensteuern (Withholding Tax) fällig werden. Denn »bei Einkünften, die aus Vergütungen für die Überlassung der Nutzung oder des Rechts auf Nutzung von Rechten, insbesondere von Urheberrechten und gewerblichen Schutzrechten […]«, kann das Finanzamt auch ausländische Rechteinhaber versteuern. Diese Regelung gilt natürlich auch im umgekehrten Fall. Denn auch ausländische Finanzbehörden können den deutschen Entwickler für die Besteuerung heranziehen. Praktisch wird die Steuer aber selten von dem Lizenzgeber (Entwickler) entrichtet. Der inländische Verwerter (Publisher, Plattformbetreiber) behält die Quellensteuer von der Vergütung ein und führt sie an das Finanzamt ab. Die Steuer wird ab dem Zeitpunkt fällig, zu dem die Vergütungen dem Entwickler zufließen. Der Publisher hat die innerhalb eines Kalendervierteljahres einbehaltene Steuer jeweils bis zum 10. des folgenden Monats an das für ihn zuständige Finanzamt abzuführen. Wenn die Leistungen als Lizenzvergütung deklariert werden, so sind auf die Vergütungen für den Entwickler Quellensteuern einzubehalten, sollte ein Auslandssachverhalt und keine Befreiung von der Quellensteuer vorliegen. Die Auszahlungsbeträge wären dann – bei einer entsprechenden Quellensteuerklausel im Vertrag – zu kürzen.

Beseitigungsfristen

Fehler in Onlinespielen kosten viel Geld, beispielsweise, wenn die Spiele nicht erreichbar sind oder kritische Fehler dazu führen, dass die Bezahlung im Spiel nicht möglich ist. Deshalb gibt es besondere Service-Level-Agreements, die Fehlerklassen definieren und dazu Reaktionszeiten und Beseitigungszeiten vereinbaren. Üblich sind drei Kategorien: 1. Kritisch, 2. Schwerwiegend sowie 3. Unproblematisch. Die Einzelheiten habe ich bereits in Making Games 3/2010 erläutert. Dabei lässt sich definieren, wer die Hoheit darüber hat, die Mängel zu klassifizieren. Die Einordnung des Mangels wirken sich auf die Reaktionszeiten aus. Je kritischer, desto schneller muss reagiert und der Fehler behoben werden. Da es weder gesetzlich normierte noch sonstige allgemein gültige Kriterien für die Klassifizierung von Mängeln und die Beseitigungsfristen gibt, ist eine Definition im Vertrag dringend zu raten. Starre Beseitigungszeiten sind hingegen schwierig zu definieren. Auch wenn viele Vertragsmuster dazu raten, ist kaum vorhersehbar, wie viel Zeit ein Entwickler benötigt, um einen Mangel im Spiel tatsächlich zu beseitigen. Zu den Service-Level-Vereinbarungen im Einzelnen kommen wir in der nächsten Ausgabe.

Mitwirkungs- und Vermarktungspflichten des Publishers

Die Wertschöpfung bei Onlinespielen ist mehrstufig. Ein Großteil der Wertschöpfung entsteht durch das Marketing sowie durch das kontinuierliche Binden und Einwerben von Spielern. Diese Kosten sollten abzugsfähig sein. Insgesamt sollte sich auswirken, ob das Spiel nach dem Auseinandergehen der Parteien von dem Plattformbetreiber weiterbetrieben werden kann, ohne dass eine Beteiligung abzugeben ist, oder ob während der gesamten Betriebszeit eine Art Mindestlizenz zu zahlen ist. Die durch das abgezogene Marketing gewonnen Spieler verbleiben ja im »Vermögen« des Plattforminhabers und können gewinnbringend für weitere Spiele eingebunden werden. Davon sollte also abhängen, wie hoch die Marge ist und welche Kosten in welchem Umfang abgezogen werden können.

Entwickler, die auf einer starken Marke entwickeln oder eine gute Verhandlungsposition haben, oder Entwickler, die einen Großteil der Vorarbeiten selbst finanziert haben, werden Wert darauf legen, dass das Spiel kommerziell erfolgreich wird. Sie müssen deshalb in jedem Fall absichern, dass der Plattformbetreiber hinreichend viele Marketingaufwendungen unternimmt und eine kritische Anzahl von Spielern einwirbt. Er wird außerdem die Plattform und Umgebung, auf der das Spiel eingesetzt wird, definieren wollen:

> Sales Target, Cenchmarks, unilateral modification: The Parties agree that the three month average of Net Receipts shall not be lower than EUR … and the Royalties of Licensor shall amount at least EUR … per month.

Die Rechtsfolgen, die daran anknüpfen können, variieren. Zum einen kann die Vergütung als Mindestvergütung definiert werden. Die Differenz zwischen Erlösen und Zielwert hat dann der Plattformbetreiber zu tragen. Diese Gestaltungsform wird oft als Minimum-Garantie bezeichnet. Andere Verträge sehen vor, dass dann der Vertrag kündbar ist, wieder andere Verträge lassen dann die Exklusivität der Nutzungsrechte entfallen.

> If one of these Benchmarks are not achieved until dd/mm/yyyy or during a subsequent term of at least three month, Licensor shall be entitled modify the agreement to a non-exclusive license, the aforementioned royalty shall then also be increased by 10%.

Mischmodelle verlangen eine Minimum-Garantie und erlauben es dem Publisher, diese zu verrechnen – allerdings nur dann, wenn eine monatliche Mindestsumme ausgeschüttet wird. Ein Teil der Mindestausschüttung ist dann in der Regel für die Weiterentwicklung und Wartung des Spiels zu verwenden.

> Publisher shall recoup the Minimum Guarantee at the royalty rate set forth below from once the monthly Net Receipts exceed EUR … provided that a minimum monthly royalty of EUR … shall be paid to Developer.

In der nächsten Ausgabe gehen wir genauer auf Service Level Agreements und den Umgang mit Ausfällen des Spiels während des Betriebs ein.

Claas Oehler

Von der Vision zum fertigen Spiel
Post Mortem Gargoyles

Innerhalb eines Semesters entstand an der Games Academy der Third-Person-Shooter Gargoyles mit der Unreal Engine 3. Wie die Studenten ihre Vision verwirklicht, technische Hürden genommen und das Team in dieser Zeit zusammengehalten haben, erzählen sie uns im Post Mortem.

Marvin Augenstein

ist Student an der Games Academy Berlin und war als Teamlead und Producer tätig.

Josef Vorbeck

ist Student an der Games Academy Berlin und war ebenfalls für das Producing zuständig.

Friedrich Brilling

ist Student an der Games Academy Berlin und war für das Game Design verantwortlich.

Svenja Hildebrandt

ist Studentin an der Games Academy Berlin und kümmerte sich um 3D Art und Animation.

Marko Plavsic

ist Student an der Games Academy Berlin und war als Programmierer am Projekt beteiligt.

Am Anfang unseres Studentenprojekts »eyecandy« stand die Zielsetzung: »eine Spielidee, die Spaß verspricht, umsetzbar ist und unseren grafischen Visionen gerecht wird. Wir, zwölf Studenten der Games Academy Berlin, wollten diese Idee innerhalb von vier Monaten in ein spielbares Stück Software verwandeln, und das unter realen Entwicklungsbedingungen. Natürlich keine einfache Aufgabe, zumal sich die meisten von uns erst im zweiten Semester befanden und noch nicht in allen Bereichen Erfahrungswerte sammeln konnten. Zunächst ging es um die entscheidende Frage, welche Vision wir eigentlich umsetzen und welche Emotionen wir transportieren wollten. Einer unserer Game Designer hatte bereits vorsorglich einen rudimentären Prototyp entsprechend seiner Vorstellung gebaut. In diesem sprang der Spieler als kleiner Gargoyle über Dächer, bewarf andere Gargoyles mit Ziegelsteinen und konnte sich zu Täuschungszwecken in Stein verwandeln. Das ganze Team war begeistert von der einfachen, aber doch originellen Idee. Darüber hinaus wollten wir das Abenteuer in einem viktorianischen Setting ansiedeln, in dem der Spieler aus Third-Person-Sicht seinen Gargoyle steuern und gegen andere Mitspieler antreten lassen konnte. Uns war sehr wichtig, beim Spieler das Gefühl der Schadenfreude zu erzeugen. Der Gargoyle sollte also eine kleine verschlagene Kreatur sein und nichts lieber im Sinn haben, als seine Konkurrenten vom Dach zu befördern oder ihnen zumindest den ein oder anderen Ziegelstein an den Kopf zu hauen. Weil Gargoyles je-

Das ambitionierte Studentenprojekt »Gargoyles – Stoned till Dawn« wurde innerhalb von nur vier Monaten und von zwölf Teammitgliedern an der Games Academy Berlin entwickelt. Als technische Grundlage diente das **Unreal Development Kit** von Epic.

doch nicht ausschließlich für das Werfen von Ziegeln bekannt sind, sollten sie zusätzlich noch Wasser speien können.

Analyse und Koordination

Durch eine Kosten-Nutzen-Analyse in der zweiten Woche identifizierten wir die entscheidenden Features für das Gelingen unserer Vision und entwickelten auf Basis dieser Analyse eine Prioritätenliste für den Programmierungs-, Art- und Designbereich. Nach diesem Prozess standen die finale Idee und die damit verbundenen Gameplay-Features fest: Der Spieler kämpft im alten London um die Herrschaft über die Dächer. Gutes altes Shooter-Feeling gepaart mit einem gewaltfreien Spielkonzept. Auch die Versteinerungsmechanik sollte unbedingt den Weg in das fertige Spiel finden, damit dem nächtlichen Treiben mehr taktische Tiefe verliehen werden konnte. Um die Vision abzurunden, setzten wir uns ein weiteres Ziel: Wir wollten eine warme, nächtliche Stadtatmosphäre erschaffen, die mit einem flüssigen Tag/Nacht-Wechsel das malerische Setting unterstreichen sollte. Natürlich lassen sich solche ehrgeizigen Ziele, noch dazu mit einem großen Studententeam, nur mit einem strukturierten Projektplan bewältigen. Unsere Producer haben sich daher für zwei der gängigsten agilen Managementmethoden entschieden, um gemeinsam dem Berg anstehender Arbeit zu trotzen. Für das Art-Department wurde nach einem einführenden Meeting die Kanban-Methode gewählt, während die Programmierer und Designer mit einer abgewandelten Scrum-Methode arbeiteten. Weil wir alle täglich noch verschiedene Seminare besuchten und damit verbunden erst ab 17 Uhr am Projekt arbeiten konnten, entschieden wir uns dafür, die Projektfortschritte wöchentlich zu kontrollieren. Die Arbeitsprozesse konnten aber täglich anhand des Kanban- sowie des Scrum-Boards für das ganze Team sichtbar gemacht werden. Ein gemeinsamer Wochenabgleich informierte zudem jeden Einzelnen über die momentane Projektlage. Darüber hinaus wurden wöchentlich ergänzende Meetings innerhalb der einzelnen Departments durchgeführt, um etwaige Probleme oder Engpässe so früh wie möglich zu erkennen und gemeinsam die nächsten Arbeitsschritte abzugleichen.

Ein essenzieller Teil dieses Konzepts war die Tatsache, dass man sich als Team vertraute und Entscheidungen, die im Zweifel der Lead Artist oder auch der Producer treffen musste, im Anschluss vom ganzen Team getragen wurden. Natürlich hatten wir gerade in der Anfangszeit oft Probleme, die einzelnen Kommunikationswege zwischen den Departments täglich aktuell zu halten. Diese Tatsache war sicherlich der Grund dafür, dass diverse Arbeitsschritte des Öfteren neu iteriert werden mussten.

Das Gargoyle-Gefühl

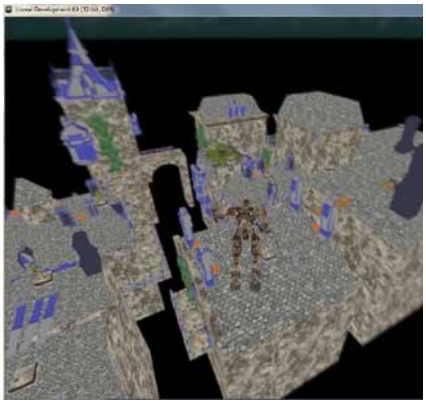
Der erste und wichtigste Schritt in der Entwicklung war es, das »Gargoyle-Gefühl« zu erwecken. Unter anderem stellte sich die Frage, wie sich wohl ein Gargoyle bewegen würde. Dabei ist nicht unbedingt die Ästhetik der Bewegung gemeint, sondern vielmehr, welchen Regeln der Avatar in der Spielwelt unterworfen ist. Wie weit kann der Gargoyle springen und gleiten? Wie schnell bewegt sich der Gargoyle im Verhältnis zur Spielwelt? Wie robust soll er sein? Diese Fragen waren uns zwar schon von Anfang an bewusst, gewannen im Laufe der Entwicklung jedoch immer mehr an Bedeutung. Besagte Parameter wurden zu Beginn mit relativ realistischen Werten versehen. Weil aber einige dieser Fragen von vorrangiger Bedeutung für das Level Design waren, mussten die Game Designer letztlich durch Rapid Prototyping in sehr minimalistischen Engine Maps die angenommenen Werte genauer definieren.

Bereits früh wurden Artworks erstellt, die das angestrebte **viktorianische Setting** und die damit verbundene düstere Stimmung vermitteln sollten.

Weitere Balancing-Faktoren waren etwa der Ziegelstein- sowie der Wasservorrat des Spielers, aber auch die Verteilung der Items und deren Spawn-Zeiten. Wir hatten einige Für- und-Wider-Diskussionen über die Item-Gewinnung. Eine der ursprünglichen Ideen aus dem allerersten Unity-3D-Prototyp war, dass der Spieler Schornsteine zerschlagen musste, um an Ziegelsteine zu gelangen. Aus Performancegründen und dem zusätzlichen Aufwand, ein Nahkampf-System zu implementieren, mussten wir uns von dieser Idee leider verabschieden.

Wasser als Spielelement

Auch die Regeneration des Wasserstrahls stand zur Diskussion. Eine Verteilung von Wasserquellen (z.B. Regenfässern) hätte wahrscheinlich dazu geführt, dass sich die Hatz auf den Dächern in einen sehr wässrigen Stellungskrieg verwandelt. Unsere einfache Lösung: Regen. Eine großartige Erklärung, warum sich der Wasservorrat des Spielers von allein füllt, aber trotzdem nur begrenzt einsetzbar ist. Wir hatten allerdings so



Unter anderem wurden Gameplay-Tests noch mit dem als Vorlage dienenden Charaktermodell aus **Unreal Tournament** durchgeführt. Später stellte sich heraus, dass das von den Studenten selbst gebaute Rig bestens funktionierte.

unsere Probleme mit dem Regen. Entweder war er zu künstlich und zerstörte die Immersion des Spielers, oder er war so komplex, dass er die Performance gnadenlos in die Knie zwang.

Nach einer Handvoll Iterationen kamen wir dann aber zu einer umsetzbaren Lösung, die sich auch so im aktuellen Spiel wiederfindet. All diese Elemente von Bewegung, Stats, Items, Regeneration und Spawn-Zeiten hatten in erster Linie ein Ziel: festzulegen, wie lang ein Spiel von Gargoyles dauern würde. Um den Spielern einen Anhaltspunkt zu vermitteln, bauten wir neben dem Timer auch den Tag/Nacht-Zyklus ein. Allein durch ihn konnten wir unheimlich viel Atmosphäre im Level schaffen. Außerdem sorgten wir dafür, dass subtile Elemente wie das Rascheln von Blättern, das Platschen der Gargoyle-Füße in Pfützen und das Gurren und Wegflattern der Tauben die Immersion des Spielers weiter unterstützten.

Die Entscheidung für Unreal

Natürlich bedarf es neben der Idee und einem ausreichend großen Team auch einer Technologie, die den Ansprüchen des »Developer-Alltags« gewachsen ist. Um das Spiel auf einem grafisch hohen Niveau (sowohl künstlerisch als auch technologisch) umsetzen zu können, ohne dem rar besetzten Programming-Department zu viel abzuverlangen, entschieden wir uns für das Unreal Development Kit. Das UDK wird in monatlichen Intervallen aktualisiert und kann auf eine relativ lange Existenz zurückblicken, die ihm den Aufbau einer riesigen Community und Knowledgebase ermöglichte. Es bedurfte anfangs einer gewissen Finesse, dem Funktionsumfang und der Informationsflut Herr zu werden. Das zwang uns an einigen Stellen dazu, Kompromisse einzugehen. Es blieb zum Beispiel gegen Ende nur bedingt Zeit, sich in das von der UDK gestellte »Profiling-System« einzuarbeiten, was unter anderem dazu führte, dass unser endgültiger Build nicht mit vollen 60 FPS auf der angestrebten Hardwarespezifikation lief. Die komplette Menü- und Interface-Logik wurde von den Game Designern übernommen, damit sich die Programmierer auf das Implementieren der Core-Mechaniken konzentrieren konnten.

Glücklicherweise trafen wir frühzeitig die Entscheidung, auf ein Version-Control-System (in unserem Fall Perforce) zu setzen, was uns mehrfach vor unnötigen Verzögerungen bewahrte. Auch wenn die Einarbeitung in das UDK zunächst mühsam schien, so wussten wir später in der Produktion dessen Richtlinien, Ansätze sowie Einschränkungen zu schätzen, weil dadurch die Optimierung und eine gewisse Robustheit gewährleistet wurden. Das Programmieren im UDK fand hauptsächlich mit der hauseigenen Skriptsprache UnrealC statt, was viel Durchforsten von Fremdcode und nur beschränkten bis keinen Zugang zu den in C++ nativ programmierten Klassen bedeutete. Dies wurde jedoch recht wenig vermisst, denn UnrealC ließ für das Programmieren von Game-Logik keine Wünsche offen. Im Gegenteil: Die Beschränkung zwang uns sogar dazu, uns stärker mit der Frage auseinanderzusetzen, welche Methode bzw. welches Design für gewisse Features am besten geeignet ist.

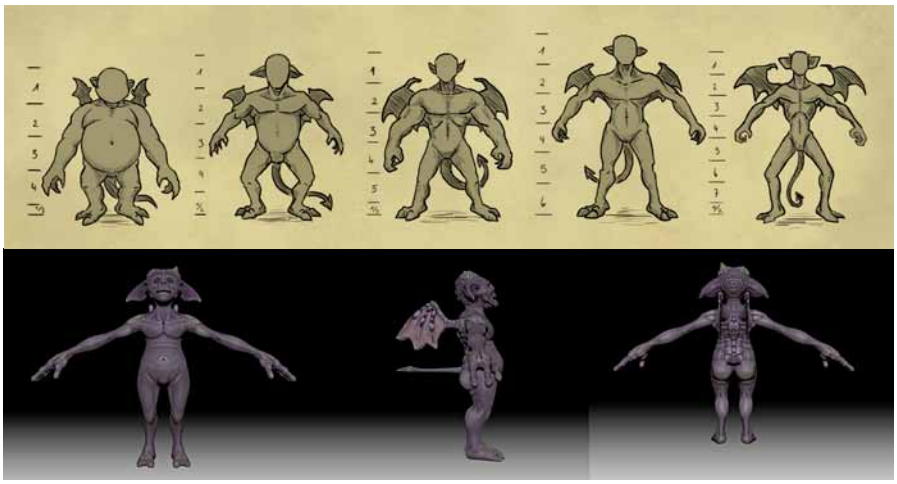
Herausforderung Netcode

Am anspruchsvollsten für die Programmierung stellte sich die Umsetzung der Netzwerkfähigkeit heraus. Das Debuggen sowie das richtige Abgleichen der unterschiedlichen Charakterzustände über das Netzwerk entwickelten sich zu einer kniffligen Aufgabe. In einem Augenblick der Produktion gab es sogar einen Build, in dem der Spieler, der die Partie gehostet hatte, die Blickrichtung der anderen Spieler bestimmte.

Natürlich hat man als UDK-Nutzer keine große Alternative, als sich den vorgegebenen Grenzen und Strukturen unterzuordnen. Wenn man sich jedoch erst einmal in die Engine eingefuchst hat, lassen sich neue Herausforderungen wesentlich leichter bewältigen.

Die Vision visualisieren

Nicht zuletzt ist es unseren Artists zu verdanken, dass die Spielwelt so lebendig wurde, wie wir es uns in unserer Vision ausgemalt hatten. Die erste Aufgabe des Art-Departments beinhaltete die grundlegende Festlegung eines zur Stimmung passenden Stils. Hier war sich das Team schnell einig, dass es weder ein zu cartooniger noch ein vollkommen realistischer Stil werden sollte. Als Vorbilder dienten deshalb Thief 3 und Borderlands – die düstere, nächtliche Atmosphäre, kombiniert mit einem Outline Shader, schien vielversprechend. Letzterer sollte auch die Tiefenwirkung der dunklen Szenerie erhöhen, indem er Objekte im Vordergrund hervorhob und nach hinten hin ausblendete. Das warme Licht der Fenster auf den Dächern des alten Londons, gepaart mit der kalten Lichtstimmung einer verregneten Nacht, sorgte für einen interessanten Kontrast. Die warm beleuchteten Bereiche dienten gleichzeitig als Markierung für die zu bespielenden Flächen, während gefährliche Straßenschluchten mit Dunkelheit und Nebel



Für das **Gargoyle-Design** waren diverse Iterationen nötig. »Lustig, aber nicht zu niedlich, gemein, aber nicht zu böse sollten sie aussehen.« Im Bild ist die Entwicklung des Konzepts bis hin zum texturierten Modell zu sehen.

gekennzeichnet waren. Höchste Priorität hatte jedoch das Design der fiesen, kleinen Gargoyles, weil diese maßgeblich den Charme des Spiels bestimmen sollten. Lustig, aber nicht zu niedlich, gemein, aber nicht zu böse sollten sie aussehen, weshalb das Design viele Iterationen und Diskussionen durchlief. Um einen konsistenten Look zu erreichen, wurde vom Lead Artist für jedes Asset mindestens eine Skizze und für die wichtigsten Modelle ein detailliertes Konzept erstellt. Hierbei wurde auch die Größe der Assets in UDK-Einheiten festgelegt sowie das ungefähre Aussehen der Texturen bestimmt. Weil auf die Performance geachtet werden musste, sollten sich auf Anraten der Dozenten möglichst viele Assets Texturen teilen können.

Wie London entstand

Für eine möglichst abwechslungsreiche Szenerie bei wenig Zeitaufwand entschied sich das Team für eine modulare Bauweise: Die Bausteine der Häuser bildeten Grundformen, welche mit frei kombinierbaren Fenstern, Erkern usw. dekoriert werden sollten. So konnten die Game Designer den Level früh unabhängig von den Artists bauen und testen. Um den Producern die Planung zu erleichtern, wurde jedes Asset zu Beginn mit einer groben Zeiteinschätzung versehen, ehe es in die Kanban-Liste aufgenommen wurde. Zum Glück stellte sich heraus, dass alle Schätzungen im Vorfeld sehr großzügig gewesen waren.

Als einer der größten Risikofaktoren war das Erstellen eines eigenen Rigs eingeplant gewesen. Wir beschlossen, dass der Gargoyle die Größe eines Standard-Unreal-Tournament-Charakters haben sollte, sodass im Notfall dessen Rig und Animationsset hätten verwendet werden können. Die Skalierung der Welt wurde entsprechend angepasst, damit der Charakter auch wie gewünscht »klein« wirkte. Diese Vorsichtsmaßnahme erwies sich jedoch als überflüssig, denn unser eigenes Rig funktionierte wider Erwarten gut. Dafür traten im Laufe des Projekts jedoch eine Reihe anderer Probleme auf. So schlugen sich das Fehlen von LODs, der Regeneffekt und der offene Levelaufbau mit hoher Weitsicht stark in den FPS nieder. Weil dies allerdings erst relativ spät auffiel, konnte das Team nur durch Reduzieren von Partikeleffekten und das Entfernen einiger unwichtigerer Assets ein wenig Performance sparen.

Safety first

Durch die solide Planung fand all das den Weg ins Spiel, was uns wichtig war. Doch noch viel wichtigere Aspekte waren die Harmonie im Team und dass jedes Teammitglied Verantwortung übernehmen konnte. Dadurch gelang uns das Spiel am Ende noch runder, als wir es zu Beginn geplant hatten. Sicherlich lief nicht alles perfekt und es wurden vereinzelt auch Fehler begangen, aber wir hatten immer einen Plan B, falls Probleme mit dem finalen Build aufgetreten wären. Wir erstellten etwa in den letzten

Der **Outline Shader** sorgt neben einem Wiedererkennungswert auch für eine erhöhte Tiefenwirkung der düsteren Umgebung. Objekte im Vordergrund werden hervorgehoben und zum Hintergrund hin ausgeblendet.

Wochen der Produktion regelmäßig einen Sicherheits-Build, um im Ernstfall ein Backup zu haben. Wir hatten den Großteil der Hürden immer sehr schnell auf dem Schirm und waren stets bemüht, das Team auf Kurs zu halten. Ebenso wichtig war aber auch das Feedback der Dozenten an der Games Academy, denn abgesehen von der Projektleitung, die uns sehr wertvolles Feedback in den regelmäßigen Reviews gab, versuchten wir immer auch, sinnvolles Feedback aus den Unterrichtseinheiten im Spiel umzusetzen. Gegen Ende des Semesters war dies zwar nicht mehr so einfach möglich, denn die Produktion war schon weit vorangeschritten, aber kleine Teile wurden dennoch immer wieder umgesetzt. Die letzten Produktionswochen nutzten wir mit vielen Gameplay-Tests durch unsere Mitstudenten, allerdings war das wertvollste Feedback jenes von Menschen, die Gargoyles bislang noch nie gesehen hatten und nun zum ersten Mal spielten.

Weil die Games Academy jedes Jahr auf der Gamescom einen Stand besitzt, hatten wir die Möglichkeit, unser Spiel unter realen Bedingungen anspielen zu lassen und uns den kritischen Meinungen der Spieler zu stellen. Als wir sahen, wie gut unser Spiel ankam und welch einen Spaß so mancher mit der Hatz über die Dächer des alten Londons hatte, wussten wir, dass wir ein wirklich tolles und solides Endprodukt abgeliefert hatten. Wir hatten riesigen Spaß, »Gargoyles – Stoned till Dawn« zu entwickeln und nahmen sehr viele Erfahrungen aus diesem Projekt mit. Ein besonderer Dank gilt noch den Knights of Soundtrack für die einzigartige Ingame-Musik sowie an Raphael Benjamin Meyer für die stimmige Musik im Hauptmenü und in den Credits. Vielen Dank auch an die verantwortliche Projektleitung und alle Dozenten, die uns tatkräftig bei der Realisierung unseres Spieles unterstützt haben.

Marvin Augenstein, Josef Vorbeck, Friedrich Brilling, Svenja Hildebrandt, Marko Plavsic

Team eyecandy

Marvin Augenstein Teamlead & Producing	Tom Kaschub Art Lead & Concepting
Josef Vorbeck Producing	Oliver Schuemann 3D Character Art
Friedrich Brilling Game Design	Sarah Mai Environment Art
Svenja Hildebrandt 3D Art & Animation	Attila Camdeivren Environment Art
Matthias Isenmann Game Design	Thomas Bruckschen 3D & Tech Art
Marko Plavsic Programming	Tjark Gloe 3D & Tech Art

Berufschance Spiele-Industrie
Praktikumsbörse

Sie studieren Informatik, möchten aber lieber Spiele statt Büro-Software programmieren? Sie kennen sich mit Maya aus und wollen Ihr künstlerisches Talent zum Beruf machen? Hier ist Ihre Chance: Spiele-Entwickler und Publisher aus ganz Deutschland suchen Praktikanten.

WWW
makinggames.de
● Online-Praktikumsbörse mit Suchfunktion

PLZ-BEREICH 00000

online4ever Limited
Art des Praktikums: Grafik 2D/3D, Internetmarketing, Interfacedesign, Gamesupport, PHP/SQL Programmierung
Mindestdauer: 1–6 Monate
Ansprechpartner: Matthias Gruhl, info@online4ever.biz, 0351/206387
Adresse: Manitiusstraße 6, 01067 Dresden

PLZ-BEREICH 10000

Brightside Games
Art des Praktikums: 20–40 Stunden nach Absprache
Bereich: 2D/3D Artist, Game Programmer
Mindestdauer: 6 Monate
Ansprechpartner: Johannes Giering, jobs@brightside-games.com, 0177 / 3354447
Adresse: Marie-Elisabeth-Lüders-Straße 1, 10625 Berlin

Dutyfarm – Let's play together
Art des Praktikums: Vollzeit, nach Absprache
Bereich: Programmierung (Flash, iPhone, C++), Grafik & Animation (2D/3D), Produktmanagement, QA
Mindestdauer: 6 Monate
Ansprechpartner: Fabian Masche, jobs@dutyfarm.de, 030 / 30368430
Adresse: Oranienstraße 6, 10997 Berlin

Exozet Games
Art des Praktikums: Nach Absprache
Bereich: Maya 3D Max Modelling, Animation
Mindestdauer: 6 Monate
Ansprechpartner: Bartol Ruzic, bewerbung@exozet.com
Adresse: Rotherstraße 20, 10245 Berlin

Future Interactive
Art des Praktikums: Studienbegleitend
Bereich: Java-Programmierung, C++-Programmierung
Mindestdauer: Nach Absprache
Ansprechpartner: Felix Kollmann, fkollmann@futureint.de, 030 / 810 593 36
Adresse: Gartenstraße 1, 14169 Berlin

GameDuell
Art des Praktikums: Studienbegleitend, nach Absprache
Bereich: Programmierung, Marketing, Controlling/Finanzen
Mindestdauer: 6 Monate
Ansprechpartner: Tina Haile, jobs@gameduell.de, 030 / 288 768 211
Adresse: Greifswalder Straße 212/213, 10405 Berlin

Games Academy
Art des Praktikums: Vollzeit
Bereich: GAMES Marketing/PR, Print, Ambient, Online, Social Communities, Web
Mindestdauer: 6 Monate
Ansprechpartner: Tina Pietsch, tina.pietsch@games-academy.de, Onlinebewerbung erwünscht, 030/29779120
Adresse: Rungestraße 20, 10179 Berlin

K1010 Media
Art des Praktikums: Vollzeit, nach Absprache
Bereich: Spiele-Entwicklung (Flash), PHP-Programmierer, Webdesign, Marketing
Mindestdauer: 3 Monate
Ansprechpartner: Martina Misikova, bewerbung@k1010.de, 030 / 57709936
Adresse: Erkelenzdamm 59/61, 10999 Berlin

MYTIVI – aktiv fernsehen
Art des Praktikums: Vollzeit, nach Absprache
Bereich: Entwicklung und Konzeption TV-Games (Transaktions-TV)
Mindestdauer: 3 Monate
Ansprechpartner: Daniel Schmitt, jobs@mytivi.com
Adresse: Oranienstraße 6, 10997 Berlin

Noumena Studios
Art des Praktikums: Vollzeit, nach Absprache
Bereich: Design (Level, Gameplay, Writer), Art (2D, 3D, Animation), Programming (C++)
Mindestdauer: 6 Monate
Ansprechpartner: André Schmitz, jobs@noumenastudios.com, 030 / 53148825
Adresse: Lützowstraße 33, 10785 Berlin

wooga – world of gaming
Art des Praktikums: Vollzeit
Bereich: Quality Assurance/ Testen neuer Spielversionen
Mindestdauer: 3 Monate
Ansprechpartner: Peggy Beschnitt, peggy@wooga.com
Adresse: Saarbrücker Str. 38, 10405 Berlin

Yager Development
Art des Praktikums: Vollzeit, Praktikumssemester, studienbegleitend, praktische Diplomarbeit
Bereich: Tester
Mindestdauer: 3 Monate
Ansprechpartner: Eike Stolley, eike.stolley@yager.de
Adresse: Pfuelstraße 5, 10997 Berlin

PLZ-BEREICH 20000

Bigpoint
Art des Praktikums: Vollzeit
Bereich: Produktanalyse/Produktmanagement, Gamedesign
Mindestdauer: 3 Monate
Ansprechpartner: Human Resources, personal@bigpoint.net, Betreff bitte »Praktikumsbörse, Produktanalyse« oder »Praktikumsbörse, Gamedesign«
Adresse: Christoph-Probst-Weg 3, 20251 Hamburg

Borgmeier Media Communication
Art des Praktikums: Praktikum für Studenten mit Schwerpunkt Marketing
Bereich: Games-PR, Corporate-PR, Marketing
Mindestdauer: 3 Monate
Ansprechpartner: Tanja Deilecke, deilecke@agentur-borgmeier.de, 040 / 413 09 60
Adresse: Rothenbaum Chaussee 5, 20148 Hamburg

CNT Media
Art des Praktikums: Nach Absprache (Übernahme möglich)
Bereich: Kommunikation, Level-Design, Projekt-Planung
Mindestdauer: 3 Monate
Ansprechpartner: Cornel Hillmann, ch@cnt-media.net
Adresse: Süderstraße 159a, 20537 Hamburg

Daedalic Entertainment
Art des Praktikums: Vollzeit
Bereich: Programmierung, Grafik, Game Design, Marketing,
Mindestdauer: 3 Monate, besser 6 Monate
Ansprechpartner: Steffen Boos, 040 / 43261272 sb@daedalic.de
Adresse: Papenreye 53, Workport Unit 2, 22453 Hamburg

dtp entertainment
Art des Praktikums: Nach Absprache
Bereich: PR, Marketing, Grafik, Test, Produktmanagement
Mindestdauer: 6 Monate
Ansprechpartner: Ulrike Mundt, 040 / 6699100 u.mundt@dtp-entertainment.com
Adresse: Goldbekplatz 3–5, 22303 Hamburg

Games Foundation / Studio Kiel
Art des Praktikums: Vollzeit, nach Absprache
Bereich: Programmierung, Grafik
Ansprechpartner: 0171 / 6905958, info@gamesfoundation.com
Adresse: Hansastraße 24, 24118 Kiel

Goodgame Studios / Altigi
Art des Praktikums: nach Absprache
Bereich: Online Marketing, Web-Promotion, Game Development
Mindestdauer: 3 Monate
Ansprechpartner: Hendrik Mainka, jobs@goodgamestudios.com
Adresse: Theodorstraße 42–90, Haus 11, 22761 Hamburg

Indigo Pearl
Art des Praktikums: Nach Absprache
Bereich: PR-Agentur
Mindestdauer: 6 Monate
Ansprechpartner: Torsten Oppermann, to@indigopearl.de, 040 / 253 302 67
Adresse: Richardstraße 45, 22081 Hamburg

InnoGames
Art des Praktikums: Vollzeit, nach Absprache
Bereich: Mediadesign, Grafikdesign, PR
Mindestdauer: 3 Monate
Ansprechpartner: Anna Sparke, career@innogames.de
Adresse: Harburger Schloßstraße 28, 21079 Hamburg

INPROMO
Art des Praktikums: Vollzeit
Bereich: Social Media Marketing für Entertainment-Kunden
Mindestdauer: 6 Monate
Ansprechpartner: Kerstin Saathoff, bewerbung@inpromo.de
Adresse: Banksstrasse 2b, 20097 Hamburg

northworks
Art des Praktikums: Vollzeit, nach Absprache
Bereich: Online-Marketing/Mediadesign, Grafikdesign, Systemadministration, Softwareentwicklung
Mindestdauer: 4 Monate
Ansprechpartner: Anna-Katja Sparke
Adresse: Stahltwiete 23, 22761 Hamburg

Splitscreen Studios
Art des Praktikums: Vollzeit
Bereich: Community Management, Programmierung, 3D-Grafik
Mindestdauer: 4 Monate
Ansprechpartner: Gustaf Stechmann, gus@splitscreenstudios.com
Adresse: Neuer Kamp 30, 20357 Hamburg

Spotsonfire
Art des Praktikums: Nach Absprache
Bereich: Marketing, Programmierung, 2D-Design, Illustration, Animation
Mindestdauer: 3 Monate
Ansprechpartner: jobs@spotsonfire.com, 040 / 209 346 330
Adresse: Hongkongstraße 7, 20457 Hamburg

Ticking Bomb Games
Art des Praktikums: Vollzeit
Bereich: Programmierung, Grafik
Mindestdauer: 3 bis 6 Monate
Ansprechpartner: Marco Schultz, jobs@ticking-bomb-games.de
Adresse: Böckmannstraße 15, 20099 Hamburg

wwg worldwidegames / Playnik
Art des Praktikums: Vollzeit
Bereich: Game Design, Product Management, Business Development, Community Management
Mindestdauer: 3 Monate
Ansprechpartner: Susan Kohnke, sk@playnik.com
Adresse: Richardstraße 45, 22081 Hamburg

PLZ-BEREICH 30000

Reakktor Media
Art des Praktikums: Nach Absprache
Bereich: 3D-Grafik, Web-Programmierung, Leveldesign
Mindestdauer: 6 Monate
Ansprechpartner: Kirk Lenke, linke@reakktor.com, 0511 / 540 980
Adresse: Arnswaldtstraße 10, 30159 Hannover

Rocketbox Studios
Art des Praktikums: Nach Absprache
Bereich: 3D-Grafik
Mindestdauer: 5 Monate
Ansprechpartner: Markus Wojcik, jobs@rocketbox.de, 0511 / 935 72 80
Adresse: Vahrenwalder Straße 7, 30165 Hannover

Xybris Interactive
Art des Praktikums: Nach Absprache
Bereich: Programmierung (Java, PHP), Grafik, Webdesign (CBS)
Mindestdauer: 3 Monate
Ansprechpartner: Christian Sauer, info@xybris.com
Adresse: Apfelstraße 12a, 33613 Bielefeld

PLZ-BEREICH 40000

bhv Software
Art des Praktikums: Studienbegleitend, Vollzeit
Bereich: Produktmanagement, Games, Non-Games, Edutainment, Herstellung, PR
Mindestdauer: 3 Monate
Ansprechpartner: Dieter Dedeke, bewerbung@bhv.de, 02131 / 765 01,
Adresse: Novesiastraße 60, 41564 Kaarst

Ubisoft / Blue Byte
Art des Praktikums: Nach Absprache
Bereich: Programmierung, Game Design, Level Design, Grafik, Projektmanagement
Ansprechpartner: siehe unter http//job.ubisoft.com/MAIN/careerportal, 0211 / 338 000
Adresse: Adlerstraße 74, 40211 Düsseldorf

Zone 2 media
Art des Praktikums: Vollzeit, nach Absprache
Bereich: Programmierung, 3D-Grafik
Mindestdauer: 6 Monate, nach Absprache
Ansprechpartner: Holger Offermanns, h.offermanns@zone2.de, 02150 / 705660
Adresse: Am Wasserturm 6, 40668 Meerbusch

PLZ-BEREICH 50000

Bright Future
Art des Praktikums: Vollzeit, Praxis-Semester
Bereich: C++-Programmierung
Dauer: 4 Monate, besser 6 Monate
Ansprechpartner: Gerald Köhler, jobs@brightfuture.de, 0221 / 78982171
Adresse: Im Zollhafen 15-17, 50678 Köln

EA
Art des Praktikums: Vollzeit
Bereich: PR, Marketing, Sales, Verwaltung
Mindestdauer: 4 Monate
Ansprechpartner: Volker Pinsdorf, www.electronic-arts.de, 0221 / 975820
Adresse: Im Zollhafen 15-17, 50678 Köln

EA Phenomic
Art des Praktikums: Vollzeit
Bereich: 2D/3D-Grafik, 2D-Grafik, 3D-Grafik, Programmierung, IT-Administration, Qualitätssicherung
Mindestdauer: 4 Monate
Ansprechpartner: Volker Pinsdorf, www.phenomic.de, 06132 / 78350
Adresse: Binger Straße 38, 55218 Ingelheim

Egosoft
Art des Praktikums: Studienbegleitend oder normal
Bereich: 3D-Grafik, Design, Programmierung
Dauer: Mindestens 6 Monate
Ansprechpartner: Bernd Lehahn, bernd@egosoft.com
Adresse: Heidestraße 4, 52146 Würselen

Headup Games
Art des Praktikums: Vollzeit
Bereich: PR, Marketing, und Producing
Dauer: 6 Monate
Ansprechpartner: Michael Zolna, info@headupgames.com
Adresse: Nordstraße 104, 52353 Düren

Independent Arts Software
Art des Praktikums: Nach Absprache
Bereich: 3D-Grafik und Animation, vorzugsweise mit Softimage XSI, 2D-Grafik und Animation für GBA-Projekte, Tool, 3D-Programmierung unter C++
Mindestdauer: 3 Monate, besser 6 Monate
Ansprechpartner: Holger Kuchling, ceo@independent-arts-software.de, 02381 / 688 292
Adresse: Münsterstraße 5, 59065 Hamm

Related Designs
Art des Praktikums: Studienbegleitend, Vollzeit, Praktikumssemester
Bereich: Programmierung, Grafik, Game Design
Mindestdauer: 6 Monate
Ansprechpartner: Svenja Büll, jobs@related-designs.de
Adresse: Römerpassage 1, 55116 Mainz

Spacetale
Art des Praktikums: Studienbegleitend, Vollzeit
Bereich: 2D/3D-Grafik
Mindestdauer: 3 Monate
Ansprechpartner: Michael Wolf, jobs@spacetale.de
Adresse: Boppstraße 26, 55118 Mainz

PLZ-BEREICH 60000

bitComposer Games
Art des Praktikums: Nach Absprache
Bereich: Marketing & PR
Mindestdauer: 3 Monate
Ansprechpartner: Lukasz Szczepanczyk, info@bit-Composer.com, 06196 / 7793810
Adresse: Mergenthalerallee 79-81, 65760 Eschborn

Cooee GmbH
Art des Praktikums: Nach Absprache
Bereich: QA, Grafik, Webprogrammierung
Mindestdauer: 3 Monate
Ansprechpartner: Boris Karklins, jobs@clubcooee.com, 06301 / 703 290
Adresse: Opelstraße 10, 67661 Kaiserslautern

Deck 13
Art des Praktikums: Nach Absprache
Bereich: 2D/3D-Design, 3D Character Artist/Animator, Concept Art, PR-Management, Webdesign
Mindestdauer: 6 Monate
Ansprechpartner: Sarah Wrensch, jobs@deck13.de
Adresse: Baselerstraße 46, 60329 Frankfurt

Dutyfarm
Art des Praktikums: Studienbegleitend, Praxis-Semester
Bereich: Programmierung (Flash Actionscript, PHP, JSP, Datenbankprogrammierung), Grafik & Gamedesign (2D/3D), Produktmanagement, PR, QA
Mindestdauer: 6 Monate
Ansprechpartner: Daniel Schmitt, jobs@dutyfarm.com
Adresse: Oranienstraße 6, 10997 Berlin

extra toxic
Art des Praktikums: Studienbegleitend, nach Absprache
Bereich: Programmierung
Mindestdauer: 3 Monate
Ansprechpartner: Ivica Aracic, jobs@extratoxic.com, 069/ 40897143
Adresse: Hanauer Landstraße 196a, 60314 Frankfurt

Kalypso Media
Art des Praktikums: Vollzeit, nach Absprache
Bereich: Community- und Forenbetreuung, Testing, Back-Office-Support
Mindestdauer: 4 bis 6 Monate
Ansprechpartner: Anika Thun, jobs@kalypsomedia.com
Adresse: Wilhelm-Leuschner-Straße 11-13, 67547 Worms

Moonbyte Games
Art des Praktikums: Vollzeit, nach Absprache
Bereich: Programmierung, Grafik (2D), Game Design
Mindestdauer: 3 bis 6 Monate
Ansprechpartner: Robert Clemens, hr@moonbytegames.com
Adresse: Wormser Landstraße 17, 67346 Speyer

World Cyber Games
Art des Praktikums: Nach Absprache
Bereich: PR
Mindestdauer: 3 Monate
Ansprechpartner: Thomas von Treichel, tvt@ac-entertainment.com, 06152 / 985482
Adresse: Im Neugrund 6-13, 64521 Groß Gerau

PLZ-BEREICH 70000

Feenix
Art des Praktikums: Vollzeit, Praxis-Semester
Bereich: Programmierung (Java, PHP)
Mindestdauer: 6 Monate
Ansprechpartner: Uwe Spitzmüller, uwe.spitzmueller@feenix.de
Adresse: Durmersheimerstraße 55, 76185 Karlsruhe

Filmakademie Baden-Württemberg
Art des Praktikums: Studienbegleitend, Vollzeit
Bereich: Character Design, Engine Entwicklung, Shader Development, Cell BE Programmierung, Maya Api Tool Entwicklung
Mindestdauer: 6 Monate
Ansprechpartner: Volker Helzle, research@filmakademie.de
Adresse: Mathildenstraße 20, 71638 Ludwigsburg

Gameforge
Art des Praktikums: Studienbegleitend, Vollzeit
Bereich: Grafik-Design, Web-Design, Programmierung PHP, 3D-Modelling, Game Design, Community Management
Mindestdauer: 3 bis 6 Monate
Ansprechpartner: Anett Graf, jobs@gameforge.de
Adresse: Albert-Nestler-Straße 8, 76131 Karlsruhe

Korion Simulation
Art des Praktikums: studienbegleitend, Praxis-Semester, Diplom, Bachelor
Bereich: Flash-Entwicklung, AS3-Programmierung, Webdesign, 3ds Max
Mindestdauer: 3 Monate
Ansprechpartner: Oliver Korn, oliver.korn@korion.de
Adresse: Mömpelgardstraße 16, 71640 Ludwigsburg

kr3m. media
Art des Praktikums: Nach Absprache
Bereich: Illustration, Gestaltung, Konzeption von Online-Spielen, Social Media, Webdesign, Projektmanagement, Programmierung von Flash-Games, PHP/HTML/XML
Mindestdauer: mindestens 3 Monate
Ansprechpartner: Patricia Kuban, patricia.kuban@kr3m.com, 0175 / 5635164
Adresse: Kaiserstraße 158, 76133 Karlsruhe

Sixteen Tons Entertainment
Art des Praktikums: Vollzeit, Studienbegleitend
Bereich: 2D/3D-Grafik, C++ Programmierung, Leveldesign
Mindestdauer: 6 Monate
Ansprechpartner: jobs@sixteen-tons.de
Adresse: Karlstraße 3, 72072 Tübingen

Trinigy
Art des Praktikums: Vollzeit, Praxis-Semester
Bereich: Softwareentwicklung im 3D-Game-Engine-Umfeld
Mindestdauer: 3 Monate
Ansprechpartner: Felix Röken, felix.roeken@trinigy.de, 07121 / 986993
Adresse: INKA-Businesspark, Arbachtalstr. 6, 72800 Eningen

B-Alive
Art des Praktikums: Nach Absprache
Bereich: Datenmanagement, Gamedesign und assistierende Projektleitung, 3D-Grafik, Software-Entwicklung C++
Mindestdauer: 6 Monate
Ansprechpartner: Arndt Schlichtig, jobs@b-alive.de
Adresse: Biberacher Straße 38, 88444 Ummendorf

CreateCtrl
Art des Praktikums: Vollzeit oder Teilzeit, Praxissemester, Übernahme möglich
Bereich: Entwicklung von Multiplayer-Spielen mit Flash, Flex und C# sowie Portalentwicklung mit JBuilder
Mindestdauer: 1 bis 6 Monate
Ansprechpartner: Klaus Herrmann, klaus.herrmann@createctrl.com, 089 / 55277736
Adresse: Bayerstraße 85a, 80335 München

Disney Interactive
Art des Praktikums: Vollzeit
Bereich: Marketing
Mindestdauer: 6 Monate
Ansprechpartner: Christina Ettelbrück, christina.ettelbrueck@disney.com
Adresse: Kronstadter Straße 9, 81677 München

Marchsreiter Communications
Art des Praktikums: Vollzeit
Bereich: PR
Mindestdauer: 3 Monate
Ansprechpartner: Dieter Marchsreiter, dm@marchsreiter.com
Adresse: Guldeinstraße 41a, 80339 München

Piece of Pie Studios
Art des Praktikums: nach Absprache
Bereich: Game Design/Level Design
Mindestdauer: 2 Monate
Ansprechpartner: Manfred Pack, mpack@pieceofpiestudios.com, 089 / 210205733
Adresse: Sendlinger-Tor-Platz 6, 80336 München

ProSiebenSat.1 Digital
Art des Praktikums: Vollzeit
Bereich: Marketing, Content Management und Product Management
Mindestdauer: 6 Monate
Ansprechpartner: http://prosiebensat1-jobs.com
Adresse: Medienallee 6, 85774 Unterföhring

Realmforge Studios
Art des Praktikums: Nach Absprache
Bereich: Programmierung
Mindestdauer: 6 Monate
Ansprechpartner: Christian Wolfertstetter, christian.wolfertstetter@realmforgestudios.com
Adresse: Hofer Straße 15, 81737 München

Remote Control Productions
Art des Praktikums: Vollzeit
Bereich: Game Design, Produktmanagement, Projektmanagement, Producing
Mindestdauer: 6 Monate
Ansprechpartner: Johannes Lesser, jobs@r-control.de
Adresse: Sendlinger-Tor-Platz 6, 80336 München

Take 2 Interactive
Art des Praktikums: Vollzeit
Bereich: Marketing, Produktmanagement
Mindestdauer: 6 Monate
Ansprechpartner: Jochen Till, jochen.till@take2.de, 089 / 27822100
Adresse: Agnesstraße 14, 80798 München

Travian Games
Art des Praktikums: Games Analyse & Evaluation
Bereich: Publishing
Mindestdauer: 6 Monate
Ansprechpartner: Christiane Nißl, jobs@traviangames.com
Adresse: Wilhelm-Wagenfeld-Straße 22, 80807 München

PLZ-BEREICH 90000

HandyGames
Art des Praktikums: Studienbegleitend, Vollzeit
Bereich: Grafik, Programmierung, Game- & Leveldesign
Mindestdauer: 6 Monate
Ansprechpartner: Markus Kassulke, jobs@handy-games.com, 09334 / 97570
Adresse: i_Park Klingholz 13, 97232 Giebelstadt

Jochen Winkler
Art des Praktikums: Nach Absprache
Bereich: Grafik, Programmierung.
Mindestdauer: 1 Monat
Ansprechpartner: Jochen Winkler, jobs@jochenwinkler.com
Adresse: Kronacher Straße 41, 96052 Bamberg

Pixel Natives
Art des Praktikums: Nach Absprache
Bereich: Programmierung(C#, Java),Webprogrammierung
Mindestdauer: 3 Monate, besser 6 Monate
Ansprechpartner: Wolfgang Kurz, w.kurz@buiss-ultimo.de
Adresse: Clermont-Ferrand-Allee 36, 93049 Regensburg

qforge
Art des Praktikums: Vollzeit
Bereich: Konzeption, Programmierung (Flash/Actionscript), Systemintegration/Servermanagement (Debian Linux)
Mindestdauer: 3 Monate
Ansprechpartner: Martin Schiele, M.Schiele@qforge.de
Adresse: Südwestpark 37-41, 90449 Nürnberg

Upjers
Art des Praktikums: Vollzeit, nach Absprache
Bereich: 3D-Grafik, Programmierung, PR
Mindestdauer: 6 Wochen
Ansprechpartner: Marika Schmitt, m.schmitt@upjers.com
Adresse: Seehofstraße 46, 96117 Memmelsdorf

A-1000

Sproing
Art des Praktikums: Vollzeit, nach Absprache
Bereich: Quality Assurance, Production
Mindestdauer: 5 Monate
Ansprechpartner: Johanna Schober, jobs@sproing.com, +43 / (0)1 / 604302838
Adresse: Fernkorngasse 10, 1100 Wien, Österreich

Making Games Magazin
Der Backstage-Pass der Spielebranche!

Making Games Magazin
Das Magazin für Spieleentwicklung und Business-Development

/PROFI-WISSEN
Von Insidern für Insider

/PROFI-CHANCEN
Mit Firmenregister und Praktikumsbörse

Jetzt 1x gratis testen!

Gratis-Ausgabe jetzt online bestellen:
www.makinggames.de/gratis

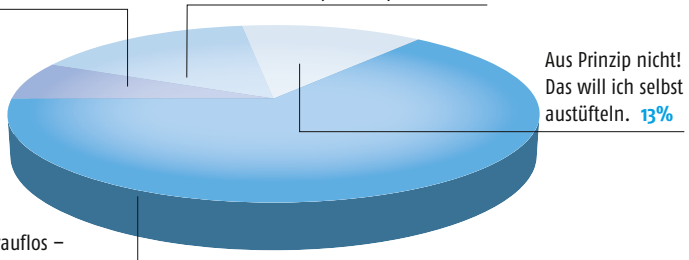
oder Tel.: 0180/5727252-275*
Mo-Fr 8-20 Uhr

Making Games Magazin erscheint im Verlag IDG Entertainment Media GmbH, Lyonel-Feininger-Str. 26, 80807 München, Handelsregister München, HRB 116413, Geschäftsführer: York von Heimburg. Die Kundenbetreuung erfolgt durch den Making Games Magazin Kundenservice, ZENIT Pressevertrieb GmbH, Postfach 810580, 70522 Stuttgart, Geschäftsführer: Joachim John, Tel.: 0180/5727252-275*, eMail: shop@makinggames.de (*aus dem dt. Festnetz nur 0,14 € pro Min., Mobilfunkpreise max. 0,42 € pro Min.)

Ausgewählte Umfragen
Dataflow

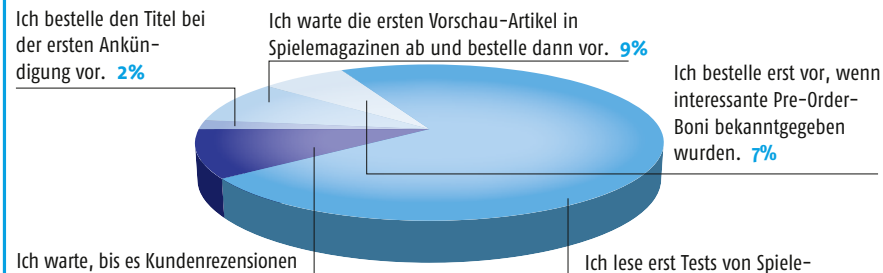
Die Redaktionen von GamePro und GameStar machen monatlich zahlreiche Umfragen unter Heftkäufern und Homepage-Usern. Eine kleine Auswahl finden Sie auf dieser Seite.

Benutzen Sie FAQs oder spezielle Skill- oder Klassen-Guides?


Ja, weil ich Angst habe, mich zu verskillen. 7%
Nur bei sehr komplexen Spielen. 16%
Aus Prinzip nicht! Das will ich selbst austüfteln. 13%
Ich spiele einfach drauflos – wird schon schiefgehen. 64%

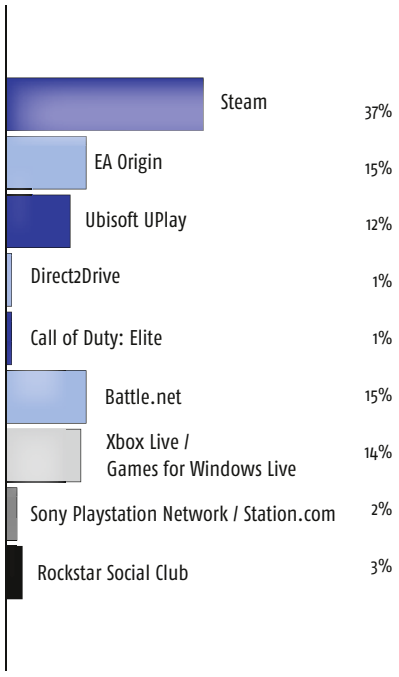
Quelle: Umfrage auf GameStar.de, 31.843 Teilnehmer

Wann treffen Sie die Kaufentscheidung für ein Spiel?


Ich bestelle den Titel bei der ersten Ankündigung vor. 2%
Ich warte die ersten Vorschau-Artikel in Spielemagazinen ab und bestelle dann vor. 9%
Ich bestelle erst vor, wenn interessante Pre-Order-Boni bekanntgegeben wurden. 7%
Ich warte, bis es Kundenrezensionen gibt und schlage dann zu. 10%
Ich lese erst Tests von Spielemagazinen und kaufe es dann. 72%

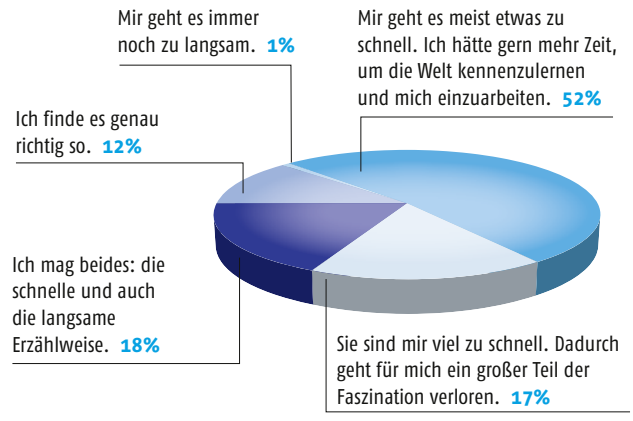
Quelle: Umfrage auf GameStar.de, 18.478 Teilnehmer

Bei welchen Online-Plattformen sind Sie angemeldet?



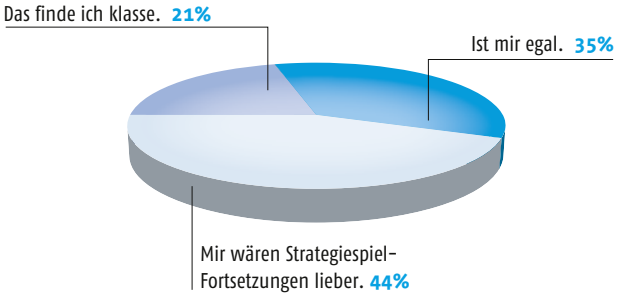
Quelle: Umfrage auf GameStar.de, 160.379 Teilnehmer

Sind Ihnen moderne Rollenspiele zu schnell oder zu langsam inszeniert?


Mir geht es immer noch zu langsam. 1%
Mir geht es meist etwas zu schnell. Ich hätte gern mehr Zeit, um die Welt kennenzulernen und mich einzuarbeiten. 52%
Ich finde es genau richtig so. 12%
Ich mag beides: die schnelle und auch die langsame Erzählweise. 18%
Sie sind mir viel zu schnell. Dadurch geht für mich ein großer Teil der Faszination verloren. 17%

Quelle: Umfrage auf GameStar.de, 26.572 Teilnehmer

Wie stehen Sie dazu, dass alte Strategieserien (z.B. Syndicate, Xcom) als Actionspiele neu aufgelegt werden?



Quelle: Redaktionelle GameStar-Umfrage, 1.109 Teilnehmer

WWW
makinggames.de
● Online-Register mit Suchfunktion. Zu jedem Key Player finden Sie zusätzlich ein detailliertes Firmenportrait.

Helfen Sie uns!

Sie haben einen Fehler entdeckt oder vermissen Ihre Firma? Dann schreiben Sie uns: **info@makinggames.de, Betreff: Firmenregister**

Die Branche im Überblick
Firmenregister

Vergessen Sie die Auskunft! Hier finden Sie sämtliche uns bekannten Publisher, Entwickler, Dienstleister, Ausbilder und Verbände der deutschsprachigen Spiele-Industrie – von Acony bis Zynga Germany.

DEVELOPMENT

Firma		Straße/Hausnummer	Postleitzahl/Ort		Telefon	E-Mail
3d-io		Rheingaustraße 94	65203	Wiesbaden	0611 / 5829602	info@3d-io.com
49Games		Waterloohain 9	22769	Hamburg		kontakt@49games.de
Acony		Mönchweilerstraße 1/8	78048	Villingen-Schwenn.	07721 / 9983829	info@aconygames.com
Aitainment		Konsul-Smidt-Straße 8f	28217	Bremen	0421 / 6195640	info@aitainment.de
Adrastea		Bielefelder Str. 5	90425	Nürnberg	0700 / 42300000	info@adrastea.com
Altigi		Theodorstr. 42-90, Haus 11	22761	Hamburg	040 / 85399366	info@altigi.de
ANIMA Entertainment		Konsul-Smidt-Straße 84	28217	Bremen	0421 / 5784220	info@anima-entertainment.de
Avaloop IT Solutions		Hütteldorferstraße 253	A-1140	Wien		office@avaloop.com
B-Alive		Biberacher Straße 38	88444	Ummendorf	07351 / 441307	admin@b-alive.de
BigBlackBlock Gamestudio		Hohenbudberger Straße 133	47229	Duisburg	02065 / 9609747	info@big-black-block.com
BIGPOINT		Christoph-Probst-Weg 3	20251	Hamburg	040 / 8814130	info@bigpoint.net
binary madness		Bundesallee 55	10715	Berlin	030 / 25093746	info@binarymadness.de
Bit Barons c/o Mediadesign Hochschule		Claudius-Keller-Straße 7	81669	München	089 / 41617744	info@bitbarons.com
Bitfield		Rigaer Straße 56	10247	Berlin	030 / 40045181	kontakt@bitfield.de
Blue Byte	key player	Adlerstraße 74	40211	Düsseldorf	0211 / 338000	info@bluebyte.de
BlurredVision		Glashütter Weg 80	22844	Norderstedt	040 / 53532683	info@blurredvision.de
Bongfish		Schönaugasse 8a/2	A-8010	Graz	+43 / 676 / 7822203	info@bongfish.com
Brightside Games		Marie-Elisabeth-Lüders-Straße 1	10625	Berlin	0177 / 3354447	
Broken Rules Interactive Media		Johann Strauss Gasse 20/8	A-1040	Wien		hello@brokenrul.es
Bronx Studios Entertainment		Lyoner Straße 14	60528	Frankfurt	069 / 665544486	info@bronxstudios.com
BSS web consulting		Dinglerstraße 5	45145	Essen	0201 / 5200044	info@bss-consulting.de
Buntware Games		Colmarerstrasse 9	79106	Freiburg	0761 / 3891598	contact@buntware.com
BURNS Games		Danziger Straße 118	10405	Berlin	030 / 40301763	info@burns-games.com
Bright Future		Im Zollhafen 15-17	50678	Köln	0221 / 78982171	info@brightfuture.de
ByteRockers		Schönhauser Allee 36	10435	Berlin	030 / 44318610	info@byterockers.de
Bytro Labs		Hopfenmarkt 33	20457	Hamburg	040 / 284645800	info@bytro.com
Caipirinha Games		Kiefernweg 1a	24161	Altenholz	0431 / 324540	info@caipirinhagames.de
Candygun Games		Hamburger Strasße 132	22083	Hamburg	040 / 20947878	
cerasus.media		Rosenthaler Straße 51	10178	Berlin	030 / 802089990	info@cerasus.de
Chasing Carrots		Hornbergstraße 94	70188	Stuttgart	0711 / 64519353	office@chasing-carrots.com
Chimera Entertainment		Sendlinger-Tor-Platz 6	80336	München	089 / 210205790	info@chimera-entertainment.com
CipSoft	key player	Gabelsbergerstraße 11	93047	Regensburg	0941 / 6308280	kontakt@cipsoft.com
Claw Design		Ludwigstraße 8	72474	Winterlingen	07434 / 3931	info@clawdesign.com
Cliffhanger Productions		Seidengasse 32/NB/7	A-1070	Wien		office@cliffhangerproductions.com
ClockStone STUDIO		Claudiastraße 14	6020	Innsbruck	+43 / 512 / 890377	info@clockstone.com
Coreplay	key player	Schleißheimer Straße 6-10	80333	München	089 / 57957556	info@coreplay.de
Cranberry Production		Podbielskistraße 166	30177	Hannover	0551 / 76170000	mail@cranberry-production.com
CreaTeam Software		Sünderup 6	24943	Flensburg	0461 / 64808	
Crenetic		Kreuzstraße 3	45468	Mülheim an der Ruhr	0208 / 4442142	info@crenetic.de
crew51		Saarwerdenstraße 8a	40547	Düsseldorf	0221 / 27228650	info@crew51.com
Crytek	key player	Grüneburgweg 16-18	60322	Frankfurt am Main	069 / 21977660	info@crytek.de
Daedalic Entertainment		Papenreye 53 – Workport Unit 2	22453	Hamburg	040 / 43261270	info@daedalic.de
Decane		Hallerstraße 5e	20146	Hamburg	040 / 42910947	info@decane.net
Deck13 Interactive	key player	Baseler Straße 46	60329	Frankfurt am Main	069 / 71671660	info@deck13.com
Destrax Entertainment		Merkurstraße 10	33739	Bielefeld		info@destrax.de
Digitalmindsoft		Otto-Lilienthal-Weg 10	89075	Ulm		info@digitalmindsoft.eu
dreiNET		Bergedorfer Straße 96	12623	Berlin		philip@dreinet.org
Dutyfarm		Oranienstraße 6	10997	Berlin	030 / 30368430	info@dutyfarm.com
DynamicHead		Hauptstraße 6	55283	Nierstein Schwabsburg	061 / 33572494	timo.weisse@dynamichead.com
EA Phenomic		Binger Straße 38	55218	Ingelheim	06132 / 783510	info@phenomic.net
Egosoft		Heidestraße 4	52146	Würselen	02405 / 4239970	egoinfo@egosoft.com
Eku Interactive		Saalburgstraße 104	61350	Bad Homburg	06172 / 597727	office@eku.de
Enter-Brain-Ment		Littenstraße 106/107	10179	Berlin		interesse@enter-brain-ment.com
eos interactive		Carl-Zeller Straße 16	75015	Berlin	07252 / 957757	info@eosinteractive.de
exDream		Hildesheimer Straße 5	30169	Hannover	0511 / 37077675	info@exdream.com
Exozet Games		Rotherstraße 20	10245	Berlin	030 / 2465600	games@exozet.com
extra toxic		Hanauer Landstraße 196a	60314	Frankfurt am Main	069 / 48440040	info@extratoxic.com
FAKT Software		Dufourstraße 28	04107	Leipzig	0341 / 14990564	info@fakt-software.de
Fancy Bytes		Niederstraße 5	26931	Huntorf	02151 / 3259912	biz@fancy-bytes.de
FDG Entertainment		Agnes-Pockels-Bogen 1	80992	München	089 / 18941370	info@fdgentertainment.com
Freudenreich & Tollknaepper		Paracelsusstraße 57	42283	Wuppertal	0202 / 2721048	mail@german-railroads.com
Funatics Software	key player	Essener Straße 66	46047	Oberhausen	0208 / 43927060	info@funatics.de

DEVELOPMENT

Firma	Straße/Hausnummer	Postleitzahl/Ort	Telefon	E-Mail
funworld	Photo Play Straße 1	A-4860 Lenzing	+43 / 7662 / 67850	info@photoplay.com
Fusionsphere Systems	Rosenau 8	86551 Aichach	0700 / 38746600	info@fusionsphere.com
GameArt Studio	Bismarckstraße 107	10625 Berlin		kontakt@GameArtStudio.de
GameDevelopment M. Molkenthin	Lerchenstraße 98	22767 Hamburg	040 / 46773207	gamedev@michael-molkenthin.de
GameDuell	Greifswalder Straße 212-213	10405 Berlin	030 / 288768210	impressum@gameduell.de
Gameforge	Albert-Nestler-Straße 8	76131 Karlsruhe	0721 / 3548080	info@gameforge.de
gameus	Kufsteinerstraße 40	83064 Raubling	08035 / 6689	info@gameus.de
Gaming Minds Studios	Georgstraße 6	33330 Gütersloh		contact@gamingmindsstudios.com
Gbanga	Brauerstraße 31	8004 Zürich	+41 / 43 / 5366701	matthias@gbanga.com
GIANTS Software GmbH	Rütistrasse 18	8952 Zürich-Schlieren	+41 / 44 / 5082176	mail_us@giants-software.com
Goodgame Studios	Theodorstraße 42-90, Haus 11	22761 Hamburg	040/ 85192490	pr@goodgamestudios.com
Greentube	Mariahilfer Straße 47/1/102	A-1060 Wien	+43 / 1 / 4945056	office@greentube.com
HandyGames	i_Park Klingholz 13	97232 Giebelstadt	09334 / 97570	info@handy-games.com
Independent Arts Software	Münsterstraße 5 / HAMTEC Haus 2b	59065 Hamm	02381 / 488818	info@independent-arts-software.de
Inflammables	Vangerowstraße 16/1	69115 Heidelberg	06221 / 7254864	info@inflammables.de
InnoGames	Harburger Schloßstraße 28	21079 Hamburg	040 / 78893350	info@innogames.de
Instance Four	Friederikastraße 65	44789 Bochum	0234 / 89038971	info@instancefour.com
Island Games	Vahrenwalder Straße 7	30165 Hannover	0511 / 35319898	kontakt@islandgames.de
Jo-Mei	Monumentenstraße 35, Aufg. C	10829 Berlin	030 / 81494080	info@jo-mei.com
K1010 Media	Erkelenzdamm 59-61	10999 Berlin	030 / 57709936	info@k1010.de
keen games	Dreieichstraße 59	60594 Frankfurt am Main	069 / 66058838	info@keengames.com
King Art	Martinistraße 47-49	28195 Bremen	0421 / 322760	contact@kingart.de
KORION Simulation	Mömpelgardstraße 16	71640 Ludwigsburg	0700 / 56746600	info@korion.de
kr3m.media	Kaiserstraße 158	76133 Karlsruhe	0721 / 18395918	
Kritzelkratz 3000	Dieselstraße 2-6	97082 Würzburg	0931 / 3536091	info@kritzelkratz.de
kunst-stoff	Rungestraße 22-24	10179 Berlin	030 / 23457178	info@kunst-stoff.de
Limbic Entertainment	Heinrich-Hertz-Straße 21	63225 Langen	06103 / 5019780	info@limbic-entertainment.de
Madcat Interactive Software	Prager Ring 4-12	66482 Zweibrücken	06332 / 791320	info@madcat.de
MaDe Games GbR Neetix - Wycislik	Hochstraße 34	41334 Nettetal		m.wycislik@madegames.com
marune	Wallbergstraße 16	82008 Unterhaching	089 / 66500749	info@marune.de
Master Creating	Winterhuder Weg 82	22085 Hamburg	040 / 33313780	mail@master-creating.com
Media Seasons	Liviastraße 6	4105 Leipzig	0341 / 2156752	contact@mediaseasons.com
mobivention	Gottfried-Hagen-Straße 24	51105 Köln	0221 / 6778110	
Moonbyte Games	Wormser Landstraße 17	67346 Speyer	06232 / 8774482	info@moonbytegames.com
netmin games	Philipp-Reis-Straße 6	55129 Mainz	06131 / 507896	schreiber@netmin.de
Neutron Games	Gustav-Meyer-Allee 25	13355 Berlin	030 / 46307222	kontakt@neutron-games.de
northworks	Stahltwiete 23	22761 Hamburg	040 / 78107075	info@northworks.de
Noumena Studios	Lützowstraße 33	10785 Berlin	030 / 53148825	info@noumenastudios.com
NovaTrix / Animedo	Schlüsselwiesen 19	70186 Stuttgart	0711 / 3102770	info@animedo.de
Nurogames GmbH	Schaafenstraße 25	50676 Köln	0221 / 3988040	info@nurogames.com
OG-Soft Productions	Zirbelstraße 48	86154 Augsburg		info@og-soft.de
online4ever Limited	Manitiusstraße 6	1067 Dresden	0351/ 2063875	info@online4ever.biz
Outline Development	Hauptstraße 122	57074 Siegen	0271 / 3131331	webmaster@outline-development.de
Piece of Pie Studios	Sendlinger-Tor-Platz 6	80336 München	089 / 210205780	info@pieceofpiestudios.com
Piranha Bytes	Ruhrallee 63	45138 Essen	0201 / 806720	info.nospam@piranha-bytes.com
PixelPlaying	Vorm Eickerberg 1a	42929 Wermelskirchen	02196 / 8899693	info@pixelplaying.de
pixeltamer.net	Clausewitzstraße 6	10629 Berlin	030 / 34347690	welcome@pixeltamer.net
Pixon	Jaffestraße 10	21109 Hamburg	040 / 33399430	welcome@pixon.de
Playa Games	Alstertor 9	20095 Hamburg	040 / 76793220	info@playa-games.com
Playgenic	Rosenheimer Straße 145 E	81671 München	089 / 513019710	info@playgenic.com
Playnik	Bielefelder Str. 5	90425 Nürnberg	0700 / 42300000	info@adrastea.com
playzo	Heinrich-Herz- Straße 6	64295 Darmstadt	06151 / 6673700	info@playzo.biz
plazz entertainment	Augustmauer 1	99084 Erfurt	0361 / 2169460	info@plazz-entertainment.com
Pop Rocket Games	Gasstraße 10	22761 Hamburg	040 / 822449040	
Procontis	Am Bauhof 18	64807 Dieburg	06071 / 430111	sales@procontis.de
Promotion Software	Karlstraße 3	72072 Tübingen	07071 / 91670	mail@promotion-software.de
qforge	Südwestpark 37 - 41	90449 Nürnberg	0911 / 3092097	info@qforge.de
Quadriga Games	August-Bebel-Straße 27	14482 Potsdam	0331 / 23189660	info@quadrigagames.com
Rabenauge	Ebelingstraße 17	10249 Berlin	030 / 21092345	the_contact@rabenauge.com
Rainfire Entertainment	Frohnhauserstraße 65	45127 Essen	0201 / 946134403	info@rainfire-entertainment.com
Rat King Entertainment	Robert-Franz-Ring 3	06108 Halle	0345 / 6141741	mail@ratking.de
Ravensburger Digital	Agnes-Pockels-Bogen 1	80992 München	089 / 12228180	tellme@ravensburger-digital.com
Reakktor Media	Kleine Duewelstraße 21	30171 Hannover	0511 / 540980	info@reakktor.com
Realmforge Studios	Hofer Straße 15	81737 München	089 / 55069134	info@realmforgestudios.com
Related Designs Software	Römerpassage 1	55116 Mainz	06131 / 554470	info@related-designs.de
remote control productions	Sendlinger-Tor-Platz 6	80336 München	089 / 21020570	info@r-control.de
Rough Sea Games	Werderstraße 23-25	68165 Mannheim	0621 / 3288684 5	blog@rough-sea.com
SAFKAS Development	Schulze-Delitzsch-Straße 39	04315 Leipzig		mail@safkas.de
Serious Games Solutions	August-Bebel-Straße 27	14482 Potsdam	0331 / 23189680	mail@serious-games-solutions.com
Silent Future	Vohwinkeler Straße 51-53	42329 Wuppertal	0202 / 5143480	info@silentfuture.de
SlipShift	Friedrichstraße 122	10117 Berlin	030 / 27581581	info@slipshift.net
Snowstep Development	Heinrich-Holtschneider-Weg 76	40489 Düsseldorf	0201 / 17848176	business@snowstep.com
Socialspiel	Schmalzhofgasse 26 / Top 5-6	A-1060 Wien		office@socialspiel.com
Solimedia Productions	Schlösserstraße 28	99084 Erfurt	0361 / 658550	hatmut.niemeier@solimedia.de
Spellbound	Weststraße 15	77694 Kehl am Rhein	07851 / 991660	info@spellbound.de
SpinBottle Games	Weidkamp 3	45355 Essen	0201 / 1789614	info@spinbottlegames.com
Spirit	Friedrichstraße 15	70174 Stuttgart	0711 / 96685510	weyrich@go4spirit.com

DEVELOPMENT

Firma		Straße/Hausnummer	Postleitzahl/Ort		Telefon	E-Mail
Splitscreen Studios		Neuer Kamp 30	20357	Hamburg		info@splitscreenstudios.com
Springwald Software		Alter Eistreff 36	44789	Bochum	0700 / 777464925	info@springwald.de
Sproing Interactive Media	key player	Fernkorngasse 10	A-1100	Wien	+43 / 1 / 60430280	office@sproing.com
Synetic		Auf der Haar 47	33332	Gütersloh	05241 / 4009900	synetic@synetic.de
Team Vienna Games		Kaiserstrasse 84/1/6	A-1070	Wien		office@teamviennagames.com
Ticking Bomb Games		Böckmannstraße 15	20099	Hamburg	040 / 80793499	info@ticking-bomb-games.de
Travian Games		Wilhelm-Wagenfeld-Straße 22	80807	München	089 / 3249150	mail@traviangames.com
Upjers	key player	Hafenstraße 13	96052	Bamberg	0951 / 510908100	mail@upjers.com
Vectority		Im Westenfeld 22a	44801	Bochum	0234 / 33385520	info@vectority.com
VisionaryX		Friedrichstraße 12	71101	Schönaich	07031 / 2850/ 414	contact@visionaryx.com
VIS Games and Entertainment		Am Keuschenend 49-51	50170	Kerpen	02273 / 999860	info@vis-games.de
Vulcando Games		Dasinger Straße 2	86165	Augsburg	0821 / 4550834	info@vulcando-games.de
Wolpertinger Games		Sendlinger-Tor-Platz 6	80336	München	089 / 210205700	contact@wolpertingergames.com
wooga	key player	Saarbrücker Straße 38	10405	Berlin	030 / 32505294	info@wooga.com
Xendex		Mariahilfer Strasse 176/6	A-1150	Wien	+43 / 1 / 89080090	office@xendex.com
Xybris Interactive		Neuendorfstraße 16d	16761	Hennigsdorf	03302 / 2085420	info@xybris.com
YAGER Development	key player	Pfuelstraße 5	10997	Berlin	030 / 69597660	contact@yager.de
Zeitland games & media		Königsallee 43	71638	Ludwigsburg	07141 / 5050565	bschreuder@Zeitland.com
Zeroscale		Schwedter Straße 9a	10119	Berlin	030 / 405057929	info@zeroscale.com
Zone 2 Media	key player	Am Wasserturm 6	40668	Meerbusch	02150 / 705660	info@zone2.de
Z-Software		Joseph-von-Fraunhoferstraße 20	44227	Dortmund	0231 / 9700780	contact@z-software.net
Zynga Game Germany		An der Welle 4	60322	Frankfurt		rocco@zynga.com

FREELANCER DEVELOPMENT

Firma		Straße/Hausnummer	Postleitzahl/Ort		Telefon	E-Mail
Jörg Beilschmidt (Game Design)	key player	Dorfstraße 12a	21407	Deutsch Evern	04131 / 2444952	j.beilschmidt@creatown.de
Marco Sowa (Producing, Licensing, Development, Localization)	key player	Postfach 420566	50899	Köln	0221 / 16916844	mail@marcosowa.de
Mariusz Ginel (Mediengestaltung, Webdesign, SEO)		Stettiner Straße 31	53840	Troisdorf	02241/9321556	mginel@mg-mediengestaltung.de
Martin Linnartz (3D Artist, Game Developer)		Bleichstraße 12	41460	Neuss	02131 / 3133306	
Nico Nowarra (Storytelling)		Friedbergstraße 9	14057	Berlin	030 / 31801899	nno@menintext.com
Niels Bauer Games (Game Design)		Sickingenstraße 37	79117	Freiburg	0761 / 66947	contact@nielsbauergames.com
Ralf C. Adam (Producing, Game Design)	key player	Hoffahrt 4b	63486	Bruchköbel	01525 / 1730000	ralf@tigerteam-productions.de
Thomas Obermaier (Programmierung)		Kapellenstraße 1	86869	Gutenberg	0170 / 3815144	contact@polyforming.com
Wolfgang Walk (Full Scale Producing)	key player	Karl-Leopold-Straße 6	76229	Karlsruhe	0721 / 48399963	wolfgang.walk@t-online.de

PUBLISHING/DISTRIBUTION

Firma		Straße/Hausnummer	Postleitzahl/Ort		Telefon	E-Mail
505 Games		Brunnfeld 2-6	93133	Burglengenfeld	09471 / 3088792	
Activision Blizzard Deutschland		Fraunhoferstraße 7	85737	Ismaning	089 / 9998170	info@activision.de
Aerosoft		Lindberghring 12	33142	Büren	02955 / 760310	info@aerosoft.de
Altigi		Theodorstr. 42-90, Haus 11	22761	Hamburg	040 / 85399366	info@altigi.de
Application Systems Heidelberg		Pleikartsförsterhof 4/1	69124	Heidelberg	06221 / 300002	kontakt@application-systems.de
BHV		Novesiastraße 60	41564	Kaarst	02131 / 76501	info@bhv.de
Bigpoint		Christoph-Probst-Weg 3	20251	Hamburg	040 / 8814130	info@bigpoint.net
bitComposer Games	key player	Mergenthalerallee 79-81	65760	Eschborn	06196 / 7793810	info@bit-composer.com
Burda:ic		Arabellastraße 23	81925	München		info@burda-ic.com
Capcom - CEG Interactive Entertainment		Barmbeker Straße 4b	22303	Hamburg	040 / 6965620	info@capcomeuro.de
Crimson Cow		Tarpen 40, Haus 5	22419	Hamburg	040 / 59355237	info@crimsoncow.de
Daedalic Entertainment	key player	Papenreye 53-Workport Unit 2	22453	Hamburg	040 / 43261270	info@daedalic.de
Dartmoorsoft		Postfach 6161	32647	Lemgo	05261 / 660731	info@dartmoorsoft.com
Disney Interactive Studios		Kronstadter Straße 9	81677	München	089 / 993400	info@disney.com
dtp entertainment	key player	Goldbekplatz 3-5	22303	Hamburg	040 / 6699100	info@dtp-entertainment.com
dtp young entertainment		Goldbekplatz 3-5	22303	Hamburg	040 / 6699100	info@dtp-young.com
Electronic Arts		Im Zollhafen 15-17	50678	Köln	0221 / 975820	
flashpoint	key player	Valvo Park - Haus 5c / Tarpen 40	22419	Hamburg	040 / 710060	info@flashpoint.de
Frogster Interactive Pictures	key player	Hardenbergstraße 9A	10623	Berlin	030 / 28470110	info@frogster.de
GameDuell		Greifswalder Straße 212-213	10405	Berlin	030 / 288768211	impressum@gameduell.de
Gameforge	key player	Albert-Nestler-Straße 8	76131	Karlsruhe	0721 / 3548080	info@gameforge.de
GameGenetics	key player	Alte Jakobstraße 85-86	10179	Berlin	030 / 922512769	info@gamegenetics.de
Gameloft		Charlottenstraße 68	10117	Berlin	030 / 20188429	gregory.wintgens@gameloft.com
Gamesload	key player	T-Online-Allee 1	64295	Darmstadt	06151 / 6800	presse@gamesload.de
GameTwist		Mariahilfer Straße 47/1/102	1060	Wien	+43 / 1494 / 5056	3rdparty@gametwist.com
gamigo	key player	Behringstraße 16b	22765	Hamburg	040 / 4118850	info@gamigo.com
Goodgame Studios		Theodorstraße 42-90, Haus 4a	22761	Hamburg	040 / 85192490	info@goodgamestudios.com
Halycon Media		An der Autobahn 24	23858	Reinfeld	04533 / 61090	infomail@halycon.de
Headup Games	key player	Nordstraße 104	52353	Düren	02421 / 4868700	info@headupgames.com
Infernum Productions		Uhlandstraße 175	10719	Berlin	030 / 91201090	contact@infernum.com
InnoGames	key player	Harburger Schloßstraße 28	21079	Hamburg	040 / 78893350	info@innogames.de
Intenium		Neuer Pferdemarkt 1	20359	Hamburg	040 / 25494296	info@intenium.de
Just A Game	key player	Greifswalder Straße 212	10405	Berlin	030 / 300137100	info@justagame.com
Kalypso Media Group	key player	Wilhelm-Leuschner-Straße 11-13	67547	Worms	06241 / 5061900	info@kalypsomedia.com
Koch Media	key player	Lochhamer Straße 9	82152	Planegg/München	089 / 242450	office@kochmedia.de
Konami		Berner Straße 103-105	60437	Frankfurt am Main	069 / 9855730	support@konami.de
Mail.Ru Games	key player	Ankelmannsplatz 1	20537	Hamburg	040 / 4143420	contact@mailrugames.com

PUBLISHING/DISTRIBUTION

Firma	Straße/Hausnummer	Postleitzahl/Ort	Telefon	E-Mail
Microsoft	Konrad-Zuse-Straße 1	85716 Unterschleißheim	089 / 31760	kunden@microsoft.com
Millform	Ringstraße 110	CH-4106 Therwil	+41 / 61 / 7220712	werner.sala@millform.ch
Morphicon Limited	Lindwurmstraße 88	80337 München	089 / 45235460	info@morphicon.de
MMM MultiMediaManufaktur	Ludwigstraße 33-37	60327 Frankfurt am Main	069 / 974600	info@multimediamanufaktur.com
Namco Bandai Partners	Carl-Benz-Straße 21	60386 Frankfurt am Main	069 / 24449100	de.info@namcobandaipartners.com
NBG	Brunnfeld 2-6	93133 Burglengenfeld	09471 / 70170	info@nbg-online.de
Neonga	Rosenthaler Straße 38	10178 Berlin	030 / 24628320	kontakt@neonga.com
Nintendo	Nintendo Center	63760 Grossostheim	06026 / 9500	info@nintendo.de
Nokia	Balcke-Dürr-Allee 2	40882 Ratingen	02102 / 89280	
Novitas	Frankfurter Straße 2	38122 Braunschweig	02574 / 9270	kontakt@greenpepper.de
Phenomedia	Friedrich-Ebert-Straße 14	44866 Bochum	02327 / 9970	info@phenomedia.com
Playa Games	Alstertor 9	20095 Hamburg	040 / 76793220	info@playa-games.com
Playgenic	Rosenheimer Straße 145E	81671 München	089 / 513019700	info@playgenic.com
ProSiebenSat.1 Digital	Medienallee 6	85774 Unterföhring	089 / 950710	info@sevengames.de
qforge	Südwestpark 37 - 41	90449 Nürnberg	0911 / 3092097	info@qforge.de
Ravensburger Digital	Kaflerstraße 8	81247 München	089 / 12228180	tellme@ravensburger-digital.com
redspotgames	Postfach 310464	80104 München	089 / 4036370	info@redspotgames.com
Rondomedia	Limitenstraße 64-78	41236 Mönchengladbach	02166 / 618660	info@rondomedia.de
RTL Games	Achnerstraße 1036	50858 Köln	0221 / 4560	entwickler@rtl-interactive.de
S.A.D.	Rötelbachstraße 91	89079 Ulm	07305 / 96290	kontakt@s-a-d.de
Sega	Rablstraße 24	81669 München	089 / 22848000	info@sega.de
Sony Computer Entertainment	Frankfurter Straße 233	63263 Neu-Isenburg	06102 / 4330	scee_presscentre@scee.net
Square Enix	Domstraße 17	20095 Hamburg	040 / 30633400	
Take 2	Agnesstraße 14	80798 München	089 / 278220	info@take2.de
THQ	Kimplerstraße 278	47807 Krefeld	02151 / 41890	thq@thq.de
Tipp24 Entertainment	Straßenbahnring 11	20251 Hamburg		info@t24e.de
Tivola Electronic Publishing	Oeverseestraße 10-12	22769 Hamburg	030 / 5363580	mail@tivola.de
Travian Games	Wilhelm-Wagenfeld-Straße 22	80807 München	089 / 3249150	mail@traviangames.com
Ubisoft	Adlerstraße 74	40211 Düsseldorf	0211 / 338000	info@ubisoft.de
Upjers	Hafenstraße 13	96052 Bamberg	0951 / 510908100	mail@upjers.com
Zuxxez	Rittnertstraße 36	76227 Karlsruhe	0721 / 464720	info@zuxxez.com

PUBLIC SECTOR

Firma	Straße/Hausnummer	Postleitzahl/Ort	Telefon	E-Mail
BIU	Rungestraße 18	10179 Berlin	030 / 24087790	info@biu-online.de
Cluster Audiovisuelle Medien Bayern	Sonnenstraße 21	80331 München	089 / 54460240	info@cam-bayern.de
FilmFernsehFonds Bayern	Sonnenstraße 21	80331 München	089 / 5446020	filmfoerderung@fff-bayern.de
G.A.M.E.	Alexanderstraße 5	10178 Berlin	030 / 257609186	info@game-bundesverband.de
gamearea FRM	Holzhausenstraße 22	60322 Frankfurt am Main	069 / 94419441	info@gamearea-frm.de
gamecity:Hamburg	Habichtstraße 41	22305 Hamburg	040 / 2270190	info@gamecity-hamburg.de
GAMEplaces	Hanauer Landstraße 126-128	60314 Frankfurt am Main	069 / 21236214	manuela.schiffner@gameplaces.de
GDI.Ruhr	Kreuzstraße 1	45468 Mülheim an der Ruhr	0208 / 69801520	info@games-ruhr.com
Hessen-IT	Abraham-Lincoln-Straße 38 - 42	65189 Wiesbaden	0611 / 7748481	info@hessen-it.de
medienboard berlin-brandenburg	August-Bebel-Straße 26-53	14482 Potsdam-Babelsberg	0331 / 743870	info@medienboard.de
Mitteldeutsche Medienförderung	Hainstraße 17–19	4109 Leipzig	0341 / 269870	info@mdm-online.de
nordmedia	Expo-Plaza 1	30539 Hannover	0511 / 1234560	info@nordmedia.de
Northstar Developers	Bödekerstr 84	30161 Hannover	0160 / 97880272	d.bruenig@collision-studios.com
ÖVUS	Aichhorngaße 14	A-1120 Wien	+43 / 1810 / 328912	office@ovus.at
USK	Torstraße 6	10119 Berlin	030 / 24088660	kontakt@usk.de

EDUCATION

Firma	Straße/Hausnummer	Postleitzahl/Ort	Telefon	E-Mail
Akademie für Medien	Balinger Straße 39A	70567 Stuttgart	0711 / 42094444	h.ehlers@amk-net.de
Cologne Game Lab / Fachhochschule Köln	Ubierring 40	50678 Köln	0221 / 82753095	info@colognegamelab.de
Design Schule Schwerin	Bergstraße 40	19055 Schwerin	0385 / 5559775	info@designschule-schwerin.de
FH Trier	Schneidershof	54293 Trier	0651 / 8103345	sek@informatik.fh-trier.de
FH Salzburg	Urstein Süd 1	A-5412 Puch/Hallein	+43 / 50 / 22110	press@fh-salzburg.ac.at
Games Academy Berlin	Rungestrasse 20	10179 Berlin	030 / 29779120	info@games-academy.de
Games Academy Frankfurt	Hanauer Landstraße 146	60314 Frankfurt	069 / 42696460	info@games-academy.de
HAW Hamburg	Finkenau 35	22081 Hamburg	040 / 428757662	
HTW Berlin	Wilhelminenhofstraße 75a	12459 Berlin	030 / 50190	
Macromedia Hochschule	Gollierstraße 4	80339 München	089 / 5441510	info.muc@macromedia.de
Mediadesign Hochschule	Lindenstraße 20-25	10969 Berlin	030 / 3992660	info-ber@mediadesign-fh.de
Mediadesign Hochschule	Werdener Straße 4	40227 Düsseldorf	0211 / 1793930	info-dus@mediadesign-fh.de
Mediadesign Hochschule	Claudius-Keller-Straße 7	81669 München	089 / 4506050	info-muc@mediadesign-fh.de
S4G School for Games	Gubener Straße 47	10243 Berlin	030 / 96595244	info@school4games.net
SAE Institute Berlin	Soltauer Straße18-22	13509 Berlin	030 / 43094470	berlin@sae.edu
SAE Institute Bochum	Metzstraße 23	44793 Bochum	0234 / 93451310	bochum@sae.edu
SAE Institute Hamburg	Feldstraße 66	20359 Hamburg	040 / 23688080	hamburg@sae.edu
SAE Institute Köln	Medienzentr. Ost, Carlswerkstr. 11c	51063 Köln	0221 / 9541220	koeln@sae.edu
SAE Institute München	Unterhachinger Straße 75	81737 München	089 / 89068771	muenchen.sae.edu
SRH Fachschulen	Bonhoefferstraße 1	69123 Heidelberg	06221 / 884225	it.heidelberg@fachschulen.srh.de
SRH Hochschule Heidelberg	Ludwig-Guttmann-Straße 6	69123 Heidelberg	06221 / 882790	irene.kohlmann@fh-heidelberg.de
Zürcher Hochschule der Künste (ZHdK)	Ausstellungsstraße 60	CH-8005 Zürich	+41 43 / 4464646	info.ddk@zhdk.ch

BUSINESS SERVICE

Firma	Straße/Hausnummer	Postleitzahl/Ort		Telefon	E-Mail
4-REAL INTERMEDIA (Lokalisation)	Ludwigstraße 32	63067	Offenbach	069 / 8090880	info@4-real.com
Anakan (Lokalisation)	Pfuelstraße 5	10997	Berlin	030 / 531420450	ana@anakan.de
Aruba Events	Kreuzstraße 1-3	45468	Mülheim an der Ruhr	0208 / 6982608	infos@aruba-events.de
arvato games task force	Carl-Bertelsmann-Straße 161F	33332	Gütersloh	05241 / 803074	kontakt@gamestaskforce.de
ATLAS Interactive Deutschland	Christoph Probst Weg 3	20251	Hamburg	040 / 413300185	sales@atlasinteractive.de
Auer & Smitkiewicz (Rechtsanwälte)	Karlsplatz 7	80335	München	089 / 45244350	smitkiewicz@auer-smitkiewicz.de
Dr. Behrmann & Härtel Rechtsanwaltsgesellschaft	Kantstraße 98	10627	Berlin	030 / 319984910	info@behrmannhaertel.de
Buschbaum Media & PR	Emanuel-Leutze Straße 21	40547	Düsseldorf	0211 / 5988140	presse@buschbaum-media.com
CULTURETRANSLATE	Lyoner Straße 34	60528	Frankfurt am Main	069 / 659998000	info@culturetranslate.de
Compad Company (Merchandising)	Habsburger Straße 85	79104	Freiburg	0761 / 5853667	info@com-pad.com
deal united	Landshuter Allee 10	80637	München	089 / 416157014	info@dealunited.com
Effective Media	Josef-Haumann-Straße 10	44866	Bochum	02327 / 22310	info@effetice-media.de
Frankfurter Buchmesse	Reineckstraße 3	60313	Frankfurt am Main	069 / 21020	info@book-fair.com
Friedmann Kommunikation	Schanzenstraße 36	51063	Köln	0221 / 5348710	info@fr-k.de
gamescom	Messeplatz 1	50679	Köln	0221 / 8213894	info@koelnmesse.de
Games Foundation	Neuer Kamp 30	20357	Hamburg		info@gamesfoundation.com
Games Quality	Zum Wasserwerk 15	15537	Erkner	03362 / 885688	info@games-quality.com
GAN Game Ad Net	Stresemannstraße 342	22761	Hamburg	040 / 67586750	
Ganz & Stock (Personnel Consulting)	Westhafentower, Westhafenplatz	60327	Frankfurt am Main	069 / 710456380	a.stock@ganz-stock.de
Gärtner PR	Destouchesstraße 68	80796	München	089 / 30766854	info@gaertner-pr.de
Game Developers Conference Europe	Kaiser-Wilhelm-Straße 30	12247	Berlin	030 / 34622644	carsten.kohlenbeck@ubm.com
GRAEF Rechtsanwälte	Jungfrauenthal 8	20149	Hamburg	040 / 80600090	www.graef.eu
IDG Entertainment Media	Lyonel-Feininger-Straße 26	80807	München	089 / 360860	dbhulapatna@idg.de
IGA Worldwide Ltd.	Oranienburger Straße 86a	10178	Berlin	030 / 2008900	info@ingameadvertising.com
IHDE & Partner (Rechtsberatung)	Schoenhauser Allee 10-11	10119	Berlin	030 / 44318660	berlin@onlinelaw.de
IMC (Full-Service Dienstleister)	Hauptstraße 172	51143	Köln	02203 / 949092	info@imc-online.net
Indigo Pearl (PR, Marketing)	Richardstraße 45	22081	Hamburg	040 / 25330279	pr@indigopearl.de
INPROMO	Gasstraße 6a	22761	Hamburg	040 / 23881771	kontakt@inpromo.de
Lightning PR	Stattropstraße 3	45138	Essen	0201 / 17122387	koetting@lightning-pr.de
Marchsreiter Communications	Guldeinstraße 41a	80339	München	089 / 51919942	dm@marchsreiter.com
ME Enterprises (Consulting)	Herrenhäuser Straße 24	30938	Burgwedel	05139 / 9599915	info@meenterprises.de
Osborne Clarke (Anwaltskanzlei)	Innere Kanalstraße 15	50823	Köln	0221 / 51084000	info@osborneclarke.de
Partnertrans (Lokalisation)	Leipziger Straße 28	40822	Mettmann	02104 / 172660	info@partnertrans.com
paysafecard.com Wertmarken	Am Europlatz 2	1120	Wien	+43 / 1 / 72083800	sales@paysafecard.com
Pixelworkshop / IG Computergrafik	Hirschengasse 1/26	1060	Wien	+43 / 1 / 5880118698	info@pixelworkshop.at
Play + Smile Marketing (PR, Marketing)	Am Wasserturm 6	40668	Meerbusch	02150 / 705660	info@play-and-smile.de
ProCom International LTD	Kisdorfer Weg 3b	24568	Kaltenkirchen	040 / 71669763	info@procom-international.com
Quinke Networks (PR, Marketing)	Bei den Mühren 70	20457	Hamburg	040 / 43093949	mail@quinke.com
remote control productions (Producing, Consulting)	Sendlinger-Tor-Platz 6	80336	München	089 / 210205700	info@r-control.de
Rode + Mathé (Rechtsberatung)	Henriettenweg 4	20259	Hamburg	040 / 43270432	info@rodemathe.de
ROESSLER PR (Marketing, PR, Consulting)	Walter-Leiske-Straße 2	60320	Frankfurt am Main	069 / 5144461	communicate@roesslerpr.de
SCC (Crossmedia)	Orleansstraße 5a	81669	München	089 / 33094660	info@scc-feld.de
Schanz International (Consulting)	Bgm.-Oberhettinger-Straße 2a	67146	Deidesheim	06326 / 6010	info@schanzgames.com
Schulte Riesenkampff (Rechtsberatung)	Hochstraße 49	60313	Frankfurt am Main	069 / 900266	schulte@schulte-lawyers.de
Selected Minds (Personalberatung)	Walther-Rathenau-Straße 16	64521	Groß-Gerau	06152 / 1871830	info@selected-minds.de
Sputnic Consulting	Ehrenbergerstr. 11	98693	Ilmenau	03677 7996990	media@sputnic-consulting.com
Station54 Community Management	Pankstraße 8-10	13127	Berlin	030 / 44716415	info@station54.de
swordfish pr (PR, Marketing)	Kaiserstraße 7	80801	München	089 / 60031641	ilka@swordfish-pr.de
Translation-Taxi	Beusselstraße 28	10553	Berlin	030 / 28371428	info@translation-taxi.com
Turtle Entertainment (eSport)	Siegburger Straße 189	50679	Köln	0221 / 88044490	info@turtle-entertainment.de
Two Pi Team	Gerresheimer Straße 9	40721	Hilden	02103 / 9411914	info@two-pi-team.de

FREELANCER BUSINESS SERVICE

Firma	Straße/Hausnummer	Postleitzahl/Ort		Telefon	E-Mail
Anja Heppelmann (Übersetzung)	Kramers Kamp 1a	48291	Telgte	02504 / 7399601	anja@spielworte.de
Anne-Petra Lellwitz (Marktforschung)	Fichtestrasse 65	63303	Dreieich	0160 / 2014364	info@apl-mafo.de
Arwed-Ralf Grenzbach (Consulting)	Friedrich-Ebert Anlage 18	60325	Frankfurt am Main	0171 / 6486720	arg@conflutainment.com
Falko Löffler (Autor)	Lindenstraße 8a	36355	Grebenhain-Ilbeshausen	06643 / 918577	mail@falkoloeffler.com
Jochen Hamma (Producing)	Wehinger Weg 9	78583	Boettingen	07429 / 910793	jochen.hammer@fantastic-realms.com
Kim Lange (Full Service Agency)	Landwehr 192	46045	Oberhausen	0178 / 2375418	lange@eins2design.de
Knut Brockmann (Game Design)	Seckbacher Landstraße 74	60389	Marburg/Lahn	0177 / 6313196	knut@needs-more-coffee.com
Marc Huppke (Projektmanagement)	Frankfurter Straße 5	64521	Gross-Gerau	06152 / 9614432	marc@make-projects.com
Marc Oberhäuser (Game Design, Consulting)	Berzbuirer Strasse 98	52355	Düren	02421 / 51547	marc.oberhauser@freelancer-games.com
Michael Bhatty (Entwicklung, Consulting)	Bismarckstraße 24	49076	Osnabrück	0541 / 1810660	michael@bhatty.de
Martin Deppe (Producing, Game Design)	Mozartstraße 1	97232	Giebelstadt	09336 / 979653	martin.deppe@wortspielmacher.de
Nicole Klinge (PR, Marketing, Events)	Glyzinenstraße 7c	80935	München	0173 / 7872013	nk@gconsultnet.de
Stefan Köhler (Lektor / Autor)	Am Salgenteich 34	38259	Salzgitter	0176 / 22512273	stefanclemenskoehler@web.de
Tom Putzki Consulting	Friedrich-Wilhelm-Straße 15	12103	Berlin	030 / 72010080	tp@tom-putzki-consulting.com

TECHNICAL SERVICE

Firma	Straße/Hausnummer	Postleitzahl/Ort	Telefon	E-Mail
Akamai Technologies	Betastraße 10B	85774 Unterföhring	089 / 94006308	contact-germany@akamai.com
Artificial Technology (KI-Middleware)	Dornierstraße 2	82178 Puchheim	089 / 80076124	info@artificial-technology.com
BiteTheBytes (Middleware)	St. Gallener Straße 9	36039 Fulda	0661 / 2006899	info@bitethebytes.com
B.TON Medien (Audio, Musik)	Luise-Ullrich-Straße 4	82031 Grünwald	089 / 189425490	mix@bton.de
Centroid (Motion Capturing)	Schlesische Straße 27	10997 Berlin	030 / 69807474	sven.bergmann@centroidanimation.com
Chromatrix (Mobile Gaming)	Kalkofenstraße 2	72411 Bodelshausen	07471 / 740828	info@chromatrix.com
Codecult (3D-Engine)	Kemnastraße 21a	44866 Bochum	02327 / 35474	info@codecult.com
ContenTainer (Leveldesign)	Eichelkopfstraße 17	63584 Gründau	06058 / 9178150	info@contentainer.de
Creature Factory	Graefestraße 33	10967 Berlin	030 / 26301330	info@creature-factory.com
Crytek (3D-Engine)	Grüneburgweg 16-18	60322 Frankfurt am Main	069 / 21977660	cryengine@crytek.com
definitv:ton (Audio)	Mozartstraße 35-37	50674 Köln	0221 / 8234220	mhoechsmann@definitiv-ton.de
Die Kolonie (Art, Design)	Adelheidstraße 82	65185 Wiesbaden	0611 / 5058765	contact@diekolonie.com
Dynamedion (Sounddesign)	Barbarossaring 8	55118 Mainz	06131 / 5847895	info@dynamedion.com
DynamicHead (Unity Services)	Hauptstraße 6	55283 Nierstein Schwabsburg	061 / 33572494	timo.weisse@dynamichead.com
Eye Rock Media (Design, Animation)	Essenerstraße 3	46047 Oberhausen	0208 / 82858923	contact@eyerock.de
fadeout (Audio Design)	Austraße 6	CH-4153 Reinach	+41 / 7 / 92793244	info@fadeout.ch
fatfoogoo	Mariahilferstraße 50	A-1070 Wien	+43 / 1 / 23622970	office@fatfoogoo.com
FFS Film- & Fernseh-Synchron	Poccistraße 3	80336 München	089 / 76708440	
Flow Studios	Gustav-Meyer-Allee 25	13355 Berlin	030 / 80929203	info@flow-studios.eu
Games Quality (Quality Assurance)	Zum Wasserwerk 15	15537 Erkner	03362 / 885688	info@games-quality.com
Global Producer (Production)	Hauptstraße 172	51143 Köln	02203 / 97722400	getintouch@global-producer.com
Glare Studios (Grafik)	Mansteinstraße 18	20253 Hamburg	02307 / 4388404	contact@glarestudios.de
GreenMamba-Studios (Animation)	Aloysstraße 7	48249 Dülmen	02594 / 7928217	info@greenmamba-studios.de
Hans HiScore (Audio)	Friedrich-Ebert-Anlage 11-13	60327 Frankfurt	0151 / 11616133	info@hans-hiscore.de
Intulo (3D, Animation)	Lindholz 89	31139 Hildesheim	05121 / 2944022	info@intulo.de
keuthen.net IT & Network Solutions	Neue Mainzer Straße 76-78	55129 Mainz	06131 / 9459880	info@keuthen.net
Lingoona (Loklisations-Middleware)	Hildastraße 38	68723 Plankstadt	06202 / 4095030	info@lingoona.com
Living Liquid Software	Langer Weg 15	A-6020 Innsbruck	+43 / 0 / 512263535	info@livingliquid.com
metricminds (Grafik)	Rüsselsheimer Straße 22	60326 Frankfurt am Main	069 / 7593380	info@metricminds.com
MobileBits (Mobile Engine)	Lerchenstraße 28	22767 Hamburg	040 / 33429566	info@mobilebits.de
modus werbung (PR, Marketing)	Neuer Kamp 30, Eingang C	20357 Hamburg	040 / 2380560	mail@modus-werbung.de
morro images (Grafik)	August-Bebel-Straße 27	14482 Potsdam-Babelsberg	0331 / 97996610	info@morroimages.com
Nevigo (Tools & Middleware)	Lyrenstraße 13	44866 Bochum	02327 / 8369840	info@nevigo.com
Periscope Studio (Audio)	Am Diebsteich 55	22761 Hamburg	040 / 31811767	info@periscopestudio.de
Phenomatics (XNA-Entwicklung, Consulting)	Hafenstrasse 47-51	A-4020 Linz/Donau	+43 / 732 / 90155230	office@phenomatics.com
Polyce (Grafik)	Littenstraße 106/107	10179 Berlin	0178 / 3592560	info@polyce.de
Port Plexus (Special Edition Design)	Am Varrenbruch 8	29690 Schwarmstedt	05071 / 96850	info@portplexus.com
Rabcat Computer Graphics (Grafik)	Fernkorngasse 10	A-1100 Wien	+43 / 1 / 5237425	mail@rabcat.com
Razer (Hardware)	Winterhuder Weg 82	22085 Hamburg	040 / 419299300	
Rocketbox Studios (Grafik)	Leonhardtstraße 10	30175 Hannover	0511 / 8984384	info@rocketbox.de
Simlity (Browsergame Engine)	Kahlaische Straße 1	07745 Jena	03641 / 3279684	
Spinor (3D-Engine)	Agnes-Pockels-Bogen 1	80992 München	089 / 54043980	spinor@spinor.com
The Light Works (Grafik)	Otto-Hahn-Straße 7	50997 Köln	02236 / 967322	info@thelightworks.com
ToBringAlive (Animation, Design)	Ostmarkstraße 49	48145 Münster	0251 / 392312	angerbauer@tobringalive.com
Tektroop (Support)	Vogelsangstrasse 36	CH-8133 Esslingen	+41/ 43 / 2776040	info@tektroop.ch
TON & SPOT audiodesign (Sound)	Immanuelkirchstraße 15	10405 Berlin	030 / 92129727	post@ton-und-spot.de
Toygardens Media (Grafik, Video)	Walderseestraße 54	30163 Hannover	0511 / 33659800	akunze@toygardens.com
Trinigy (3D-Engine)	Arbachtalstraße 6	72800 Eningen u.A.	07121 / 986993	info@trinigy.de
VCC Perfect Pictures (Grafik, Video)	Doormannsweg 43	20259 Hamburg	040 / 431690	hamburg@vcc.de
Versant (Data Management)	Wiesenkamp 22b	22359 Hamburg	040 / 609900	info@versant.com
Virgin Lands (Grafik, Cinematics)	Hauptstraße 5	97332 Volkach	09381 / 71780	info@virgin-lands.com
xaitment (KI-Middleware)	Fischbacherstraße 92	66287 Quierschied	06897 / 600800	info@xaitment.com
Z-Ground Illustration (Grafik)	Oeltzenstraße 17	30169 Hannover	0511 / 1696959	jz@z-ground.com
zuuka! (Audio, Lokalisation)	Christian-Pless-Straße 11-13	63069 Offenbach am Main	069 / 2475700	mail@zuuka.de

FREELANCER TECHNICAL SERVICE

Firma	Straße/Hausnummer	Postleitzahl/Ort	Telefon	E-Mail
Alexander N. Ostermann (Animator)	Breite Strasse 45	14199 Berlin	0176 / 10365759	nico@nico-ostermann.com
Andreas Adler (Audio)	Wernigeroder Weg 18	30419 Hannover	0511 / 3889163	andreas@adler-audiopictures.de
Daniel Pharos (Audio)	Golddistelanger 15	80937 München	0179 / 5210100	daniel@knightsofsoundtrack.com
Grafikbüro Werner (Grafik)	Beethovenstraße 40a	48317 Drensteinfurt	02508 / 985801	info@grafikbuero-werner.de
Maurice T. Anderson (Sounddesign)	Klosterstraße 3	63486 Bruchköbel	0176 / 26228136	contact@maurice-t-anderson.com
Olaf Bartsch (Sounddesign)	Sültstraße 62	10409 Berlin	0170 / 7743432	info@olafbartsch.com
Sabrina Heuer (Audio Production)	Bachemer Straße 191	50935 Köln	0163 / 8604004	sabrina@froschtatze.de

Was macht eigentlich ...
David Brevik?

Er hat Diablo erfunden, das Battle.net zum Leben erweckt, bei Blizzard seine Höhen gefeiert und bei Flagship mit Hellgate London die Tiefen kennengelernt. Wir sprechen mit David Brevik über die Zukunft von MMOs und seine derzeitige Arbeit bei Gazillion Entertainment.

David Brevik

ist Präsident und COO von Gazillion Entertainment.

David hat 1993 das Entwicklerstudio Condor gegründet, besser bekannt unter dem Namen Blizzard North. Als Project Lead, Design Lead und Lead Programmer hat er unter anderem an Diablo und Diablo 2 mitgewirkt. Außerdem hatte er die Idee für das Battle.net und ist für dessen Verwirklichung verantwortlich gewesen.

Nach seinem Weggang von Blizzard hat David die Flagship Studios mitgegründet, wo er an Hellgate London arbeitete. Nach der Schließung des Studios arbeitete er kurzzeitig für Turbine, die Macher von Herr der Ringe Online. 2009 nahm er seine Tätigkeit bei Gazillion Entertainment auf, wo er bis heute als Präsident und COO beschäftigt ist.

Making Games Was machst du derzeit, Dave?
David Brevik Ich bin aktuell Präsident und COO von Gazillion Entertainment. Wir haben diverse Projekte in der Pipeline, an denen wir im Moment arbeiten – alles MMOs. Ursprünglich bin ich aber zu Gazillion gekommen, um die Entwicklung von Marvel Universe zu leiten. Erst vor etwa acht Monaten wurde ich dann mit der Funktion des Präsidenten vertraut, jetzt kümmere ich mich daher um alle unsere Projekte und nicht nur um Marvel Universe. Es ist natürlich immer noch in Entwicklung, wird aber nicht mehr dieses Jahr erscheinen.

Making Games Was hältst du eigentlich von der Diablo-Beta, die gerade läuft?
David Brevik Ich denke, dass Blizzard einen guten Job macht, ich hatte eine gute Zeit beim Spielen. Es ist aber natürlich nicht dasselbe Diablo 3, das ich gemacht hätte. Ich wünsche ihnen aber nur das Beste, obwohl es auch einen leicht bitteren Beigeschmack für mich hat. Ich habe so hart daran gearbeitet, die Marke Diablo aufzubauen, hatte schon in der Highschool die Idee dazu, musste über einen langen Zeitraum diverse Leute von ihr überzeugen, wurde laufend abgewiesen, durfte am Ende meine Vision verwirklichen, es wurde ein fantastisches Produkt und dann musste ich das Ganze hinter mir und andere Leute ran lassen. Es fühlt sich merkwürdig an und ich hoffe, dass die Marke Diablo würdig weitergeführt wird.

Gazillion-COO **David Brevik** im Gespräch mit Making-Games-Chefredakteur Heiko Klinge.

Making Games Du warst einer der Erfinder des Battle.net. Was genau war damals eure Idee hinter der Multiplayer-Plattform?
David Brevik Im Grunde hat uns unser Publikum darauf gebracht. Wir haben uns bei Blizzard immer gefragt, was wir als Spieler gern hätten. Und ich wollte damals mit Freunden spielen, aber auf diese verdammten LAN-Partys verzichten. Lieber spiele ich gemütlich im Wohnzimmer, während mein Kumpel in seinem sitzt. Diese Idee war die Basis und drum herum hatten wir die Vorstellung von diversen Services, die wir einbauen wollten. Man sollte einfach auf »Multiplayer« klicken und loslegen können. Mit zwei laufenden Programmen gleichzeitig – zwar ein Pain in the Ass für die Programmierer– aber so sollte es sein. Mike O'Brien, Mitgründer von Arena Net, hat damals übrigens einen Großteil des Codes geschrieben.

Making Games Ihr habt aber damals noch nicht an ein Echtgeld-Auktionshaus gedacht, oder?
David Brevik Nein, nicht wirklich (lacht).
Es ging uns nur darum, dass man sich mit realen Menschen verbinden und mit ihnen spielen kann. Wir haben uns ursprünglich schon überlegt, ob wir Geld für das Battle.net nehmen sollten. Aber wir haben sehr schnell gemerkt, dass wir es den Spielern wirklich als kostenloses Feature an die Hand geben sollten.

Making Games Später hast du dann Blizzard verlassen und Flagship gegründet. Was hast du aus dieser Zeit mitgenommen? Es war ja schließlich eine richtige Achterbahnfahrt.
David Brevik Die Zeit bei Flagship war wirklich eine wilde Achterbahnfahrt. Unser größtes Problem war, dass wir uns mit Hellgate London einfach übernommen haben. Wir haben unsere eigene Technologie entwickelt, alles sollte auch noch in 3D sein, und es sollte einfach etwas Riesengroßes werden – am Ende war es dann eben auch ein riesengroßes Risiko.

Making Games Hältst du immer noch Kontakt zu deinen ehemaligen Flagship-Kollegen?
David Brevik Ja, absolut. Max und Erich Schaefer wohnen gerade mal sechs Blocks entfernt von mir und ich habe sie erst vor ein paar Wochen zum

David Brevik hat mit **Diablo 1 und 2** Meilensteine der Spielegeschichte erschaffen. Das von ihm mitentwickelte **Battle.net** setzt bis heute Maßstäbe in der Art und Weise, wie eine kostenlose Multiplayer-Plattform zu funktionieren hat.

Mittagessen getroffen. Wir sind also noch ständig in Kontakt und sehen uns häufig.

Making Games Wie hat sich aus deiner Sicht der MMO-Markt über die Jahre verändert? Gibt es überhaupt eine Veränderung?
David Brevik Der Markt hat sich diversifiziert. Die größte Aufmerksamkeit der Spieler und der Presse gilt wohl den MMOs, die als potenzielle WoW-Killer angepriesen werden. Wenn jemand heute MMO sagt, dann denken die meisten Leute zuerst an World of Warcraft oder Everquest oder Dark Age of Camelot oder Age of Conan. Diese Spiele ähneln sich in weiten Teilen stark. MMO ist im Grunde aber eher eine Spielweise als ein Spieltyp. Zum Beispiel würde ja niemand bezweifeln, dass Ultima Online ein MMO sei, aber die Spielerfahrung ist eine völlig andere. Ich denke, dass »MMO« in Zukunft bedeuten wird, dass viele Leute gemeinsam spielen, aber nicht zwingend mit demselben Spielstil. Nehmen wir etwa ein kleines Indie Game im Browser, eine Mischung aus Shooter und Rollenspiel, mit Hunderten Spielern gleichzeitig. Das wird die Richtung sein, in die sich der Markt bewegen wird.

Making Games Gibt es überhaupt noch einen Markt für klassische Abo-Modelle bei MMOs?
David Brevik Die werden es schwer haben. Es existiert schon noch ein Markt, am Ende geht es aber nur darum, wie gut dein Produkt ist. Die Menschen bezahlen für hervorragende Produkte. Punkt. Und trotzdem wirst du mit einem einfachen Zugang größere Erfolgschancen haben.

Making Games Wenn ich heute ein MMO-Projekt starte, worauf muss ich besonders achten?
David Brevik Neben dem Free2Play-Modell sollte man sich Cloud Gaming ansehen und überlegen, wie man es integrieren kann. Genauso sollte man Multiplattform-Gaming evaluieren. Meinen Charakter zuhause auf dem Desktop-PC, auf der Couch mit dem Laptop, auf der Arbeit, im Café, auf meinem Smartphone oder Tablet ansehen, jederzeit und überall am Spiel teilnehmen zu können, wird die nächste große Herausforderung sein.

Making Games Wird auf jeder Plattform dasselbe Spiel in angepasster Form laufen oder stellst du dir vor, dass nur bestimmte Teile auf der entsprechenden Plattform verfügbar sein werden?

»Ich denke, dass Blizzard bei **Diablo 3** einen guten Job macht, ich hatte eine gute Zeit beim Spielen der Beta. Es ist aber natürlich nicht dasselbe Diablo 3, das ich gemacht hätte. Ich wünsche ihnen aber nur das Beste, obwohl es auch einen leicht bitteren Beigeschmack für mich hat.«

David Brevik Ich spreche nicht davon, die Spiele auf unterschiedliche Plattformen zu portieren, sondern von dem Zugang zu einem Spiel. Man wird World of Warcraft nicht wirklich gut auf dem Smartphone spielen können. Aber wenn ich auf irgendeine Art und Weise vom Telefon aus mit dem Spiel interagieren könnte, wäre das fantastisch. Vielleicht angepasste Spiele, mit denen ich meinen Charakter verbessern kann.

Making Games Welche Art von MMO würdest du gern einmal spielen?
David Brevik Ich würde einfach gern MMOs sehen, die nicht die typische Formel verwenden und neue Wege gehen. Im Kern geht es ja immer darum, möglichst viele Leute im Spiel zu verbinden. Ich fände ein Konzept toll, bei dem die Spieler wirklich kooperativ handeln und etwas Gemeinsames schaffen, in etwa so wie in Minecraft. Es steckt immenses Potenzial darin, das riesige Publikum dazu zu bringen, miteinander zu arbeiten, oder gegeneinander zu kämpfen, oder irgendetwas anderes zu erleben. MMOs wird es immer geben, aber es geht darum, solch spannende Konzepte in Zukunft zum Leben zu erwecken.

Making Games Vielen Dank für das Interview, Dave!

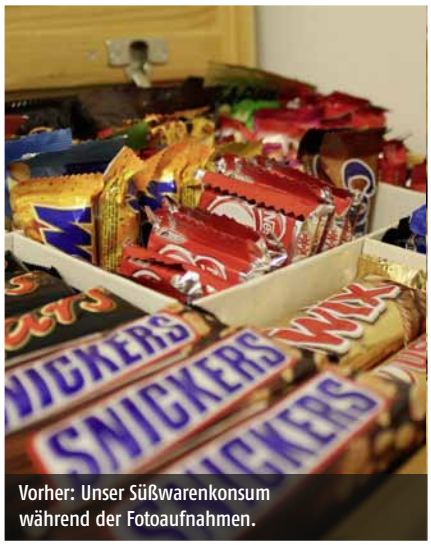
Interview: Heiko Klinge

David hat bei Gazillion Entertainment zunächst am Online-Rollenspiel **Marvel Universe** gearbeitet. Jetzt ist er als Präsident und COO für alle Spiele des Unternehmens verantwortlich.

Ein Tag bei ...
Noumena Studios

Im Herzen von Berlin, nicht unweit des Potsdamer Platzes, sitzt die Noumena.

So Charmant ist unser Empfang: Dominik Appel bei der Arbeit.



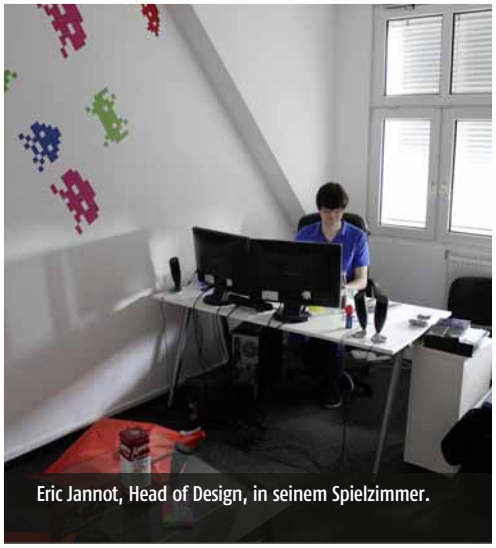
Vorher: Unser Süßwarenkonsum während der Fotoaufnahmen.

Demonicons Hauptcharakter Cairon begrüßt alle Gäste im Empfangsbereich.

Hilmar Dietrich, Head of Programming, in seinem ganz eigenen Dschungel.

Unser Kaffeekonsum – ist zwar ein Klischee, dafür aber nicht weniger wahr.

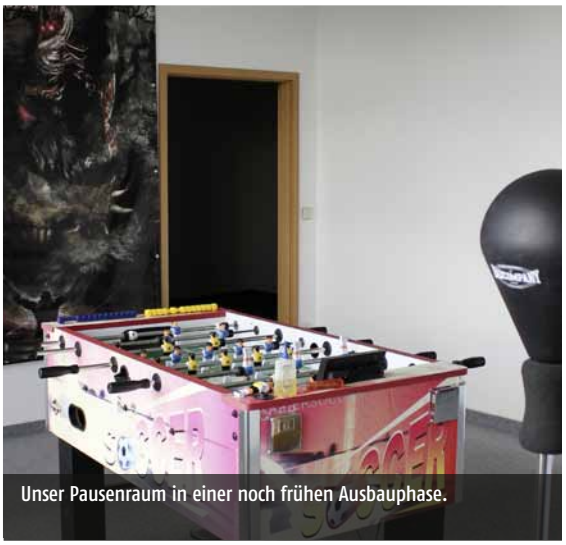
Drei Grafikerkollegen (mit farblich abgestimmten Kopfbedeckungen) bestaunen das neueste Meisterwerk unserer Concepterin Elisavet Theodosiou.


Eric Jannot, Head of Design, in seinem Spielzimmer.

Derzeit wird bei den Noumena Studios fleißig am Rollenspiel Demonicon gewerkelt. Wir zeigen indes, wie sich die Berliner über eine einzelne Bananen freuen können, was sie so alles mit Laugengebäck basteln und wo überall sonst noch Süßkram versteckt liegt.

Nachher: Zum Glück gibt's bei der Noumena noch so eine Box ...

Wenn Kreative mit Laugengebäck feiern. Künstler unbekannt, entstanden auf der letzten Firmenfeier.



Unser Pausenraum in einer noch frühen Ausbauphase.

Hier werkeln einzig und allein Programmierer mit Stil.

So lässt es sich arbeiten: Karin van der Maarel, Digital Artist Animation, auf ihrem Ball.

Warum die Animatoren David Monteiro und Alexander Rotter engumschlungen über eine Banane lachen? Lieber nicht fragen ...

Programmierer Florian Borchert lässt etwas Dampf ab.

Für eine lebensechte Animation ist uns kein Preis zu hoch: Täter Animator Alexander Rotter, Opfer Office Manager Dominik Appel.

Eine Auswahl Spielbeschreibungen von Das Schwarze Auge für unser Demonicon-Projekt.

In der nächsten Ausgabe
Making Games Vorschau

Making Games 02/2012

erscheint am 24. Februar 2012

Titelthema:

Die mobile Zukunft

Egal ob AAA-Publisher, Indie Startup oder Browsergames-Entwickler: Alle wollen sich 2012 verstärkt den mobilen Plattformen zuwenden. Fragt sich nur wie. iOS, Android oder Multiplattform? Paid Download, Freemium oder werbefinanziert? Klassisches Marketing oder User einkaufen? Wir sammeln Erfahrungswerte, Strategien und Zukunftsprognosen.

Weitere Themen:

- **Grafik**
  Making-of von Virgin Lands: So entstanden die Gebäude von ANNO 2070.
- **Programmierung**
  Unit Tests und Deployment in Unity 3D am Beispiel von Jagged Alliance Online.
- **Business**
  Wie man aus dem Nichts ein neues Unternehmen aufbaut.

IMPRESSUM

Chefredakteur
Heiko Klinge (verantwortlich, Anschrift d. Red.)

Redaktion (Freelancer)
Knut Brockmann

Lektorat (Freelancer)
Marion Schneider

Layout
Manfred Aumaier

Layout (Freelancer)
Martin Laukmanis

Redaktion
Yassin Chakhchoukh

Redaktion (Freelancer)
Martin Le

Layout (ltd.)
Sigrun Rüb

Layout
Eva Zechmeister

Freie Mitarbeiter dieser Ausgabe (Redaktion):
Sascha Ahlers, Marvin Augenstein, Friedrich Brilling, Daniel Dumont, Oliver Feige, Benedikt Grindel, Svenja Hildebrandt, Stephanie Kaiser, Prof. Dr. Christoph Klimmt, Jochen Peketz, Marko Plavsic, Dennis Raspotnig, Johannes Scharl, Christian Schmidt, Josef Vorbeck, Wolfgang Walk, Szymon Wilkosz, Jochen Winkler, Benjamin Zuckerer

Wir danken den Interview-Partnern:
David Brevik, Christoph Jennen, Kristian Segerstrale

Assistenz:
Isa Stamp, Anita Thiel, Annie Weissenberger

Korrespondenten:
Stacy Cowley (New York), Zhang Ran (Shanghai), Martyn Williams (Tokio), Peter Sayer (Paris), Sumner Lemon (Taipeh), George Dragon (Budapest), Lukas Erben (Prag)

Kundenservice

Abonnement und Einzelbestellung:
Making Games Kundenservice
ZENIT Pressevertrieb GmbH
Postfach 81 05 80, 70522 Stuttgart
Tel.: 01805 / 727252-275
Fax: 01805 / 727252-377
Österreich: Tel.: 01 / 219 55 60
Schweiz: Tel.: 071 / 314 06-15
E-Mail: shop@makinggames.de
Web: www.makinggames.de/shop
Bezugspreise:
Einzelpreis: 6,90 Euro
Jahresabo: 28,98 Euro
(beinhaltet 6 Ausgaben)
Auslandspreise u. Mengenrabatt auf Anfrage
Zahlungsmöglichkeit:
Postbank Stuttgart, BLZ 600 100 70
Konto-Nr.: 311 704
So erreichen Sie die Redaktion:
IDG Entertainment Media GmbH
Redaktion Making Games
Lyonel-Feininger-Straße 26
80807 München
Tel.: 089 / 360 86-660, Fax: 089 / 360 86-652

Anzeigenkontakt

Sales Manager Making Games (Freelancer):
Nicole Klinge, 089 / 360 86 (-680)
nklinge@idg-consultant.de
Director Sales:
Ralf Sattelberger, (-730)
Senior Sales Manager:
Kerstin Wöldering (-625)
kwoeldering@idg.de
Sales Manager:
Susanne Schreiner (-670)
sschreiner@idg.de
Sales Service Manager:
Daniela Kretschek (-691)
dkretschek@idg.de
Digitale Anzeigenannahme Print:
Manfred Aumaier (-602)
anzeigendispoprint@gamestar.de
Digitale Anzeigenannahme Online:
banner@idg.de
Anzeigenpreise:
Es gilt die Preisliste gültig ab 01.01.2010

Vertrieb Handelsauflage

MZV GmbH & Co. KG
Ohmstraße 1, 85716 Unterschleißheim
Telefon: 089 / 319 06-0, Fax: -113
E-Mail: info@mzv.de
Web: www.mzv.de

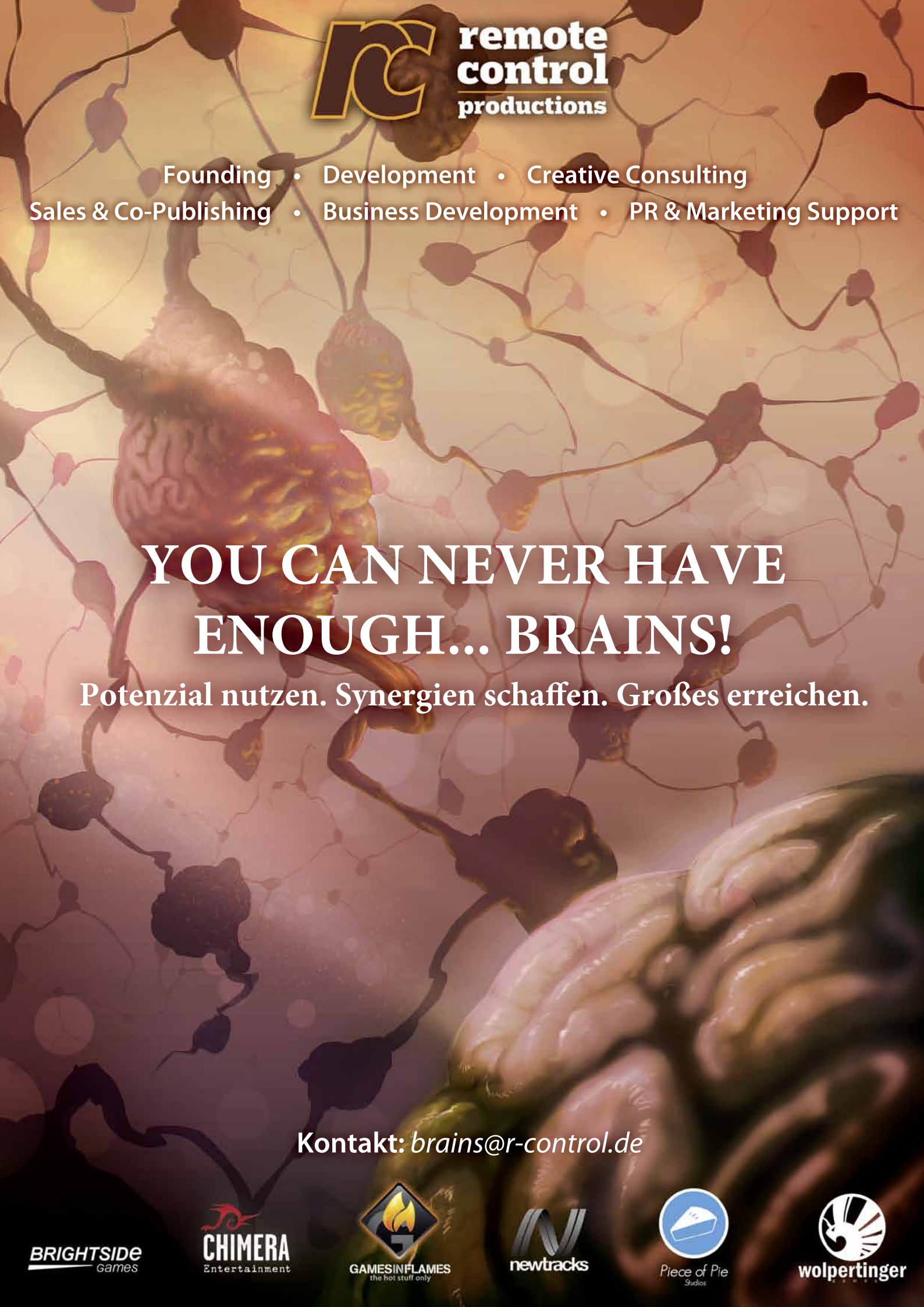
Verlag

IDG Entertainment Media GmbH
Lyonel-Feininger-Straße 26
80807 München
Tel.: 089 / 360 86-0, Fax: 089 / 360 86-501
www.idgmedia.de
Geschäftsführer: York von Heimburg
Verlagsleitung: André Horn
Leiter CRM/Marketing: Matthias Weber (-154)
Veröffentlichung gemäß §8, Absatz 3 des Gesetzes über die Presse vom 08.10.1949: Alleiniger Gesellschafter der IDG Entertainment Media GmbH ist die IDG Communications Media AG, München, die 100% Tochter der International Data Group Inc., Boston, USA, ist.
Vorstand:
York von Heimburg, Keith Arnot, Bob Carrigan
Aufsichtsratsvorsitzender:
Patrick J. McGovern
Produktionsleitung: Jutta Eckebrecht
Druck und Beilagen:
SDV Direct World GmbH
01159 Dresden

Haftung & Co.

© Copyright IDG Entertainment Media GmbH
Haftung: Für die Richtigkeit von Veröffentlichungen können Redaktion und Verlag trotz Prüfung nicht haften. Die Veröffentlichungen in Making Games erfolgen ohne Berücksichtigung eines eventuellen Patentschutzes. Auch werden Warennamen ohne Gewährleistung einer freien Anwendung benutzt.

GDC 12
learn / network / inspire
GAME DEVELOPERS CONFERENCE®
SAN FRANCISCO, CA
MARCH 5-9, 2012
EXPO DATES: MARCH 7-9
2012
WWW.GDCONF.COM
UBM
TechWeb


remote
control
productions
Founding • Development • Creative Consulting
Sales & Co-Publishing • Business Development • PR & Marketing Support
YOU CAN NEVER HAVE ENOUGH... BRAINS!
Potenzial nutzen. Synergien schaffen. Großes erreichen.
Kontakt: brains@r-control.de
BRIGHTSIDE games
CHIMERA Entertainment
GAMESINFLAMES the hot stuff only
newtracks
Piece of Pie Studios
wolpertinger